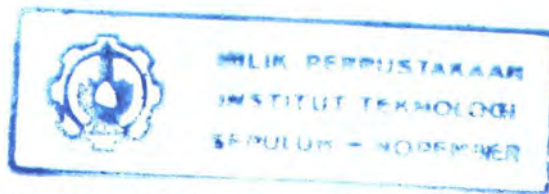


24519/H/06



Quick ASP : PEMBANGKIT KODE PROGRAM ASP UNTUK APLIKASI BASIS DATA BERBASIS WEB

TUGAS AKHIR



RSIF
006.7
Nov
2-1
2005

Disusun oleh :

Rahmi Ika Noviana
5100.100.005

PERPUSTAKAAN ITS	
Tgl. Terima	6-8-2005
Terdapat Di	H
No. Agenda Pyp.	222972

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2005

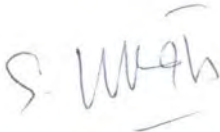
Quick ASP : PEMBANGKIT KODE PROGRAM ASP UNTUK APLIKASI BASIS DATA BERBASIS WEB

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Komputer
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya**

Mengetahui / Menyetujui,

Dosen Pembimbing I



Ir. Siti Rochimah, MT
NIP 132 103 631

Dosen Pembimbing II



Imam Kuswardayan, S.Kom
NIP 132 306 543

**SURABAYA
2005**

KATA PENGANTAR

Alhamdulillahirobbil 'alamin, segala puji bagi Allah SWT yang senantiasa memberikan kekuatan hidayah dan limpahan kasih sayang-Nya, yang memberikan kenikmatan kepada penulis untuk merasakan nikmat dan indahnya di jalan dakwah. Shalawat serta salam tetap kita curahkan kepada Rasulullah Muhammad SAW. Akhirnya penulis bisa menyelesaikan tugas akhir ini.

Penulis memberi judul pada tugas akhir ini :

PEMBANGKIT KODE PROGRAM ASP

UNTUK APLIKASI BASIS DATA BERBASIS WEB

Semoga dengan selesainya tugas akhir ini, dapat membantu dalam proses pengembangan teknologi informasi di Jurusan Teknik Informatika. Penulis menyadari bahwa tugas akhir ini jauh dari sempurna, masih banyak kekurangan-kekurangan. Oleh karena itu sangat diharapkan kritik dan saran yang membangun untuk perbaikan kedepan.

Akhirnya, penulis mengucapkan *jazakumulloh khoironkatsiro* kepada pihak – pihak yang telah membantu terselesaikannya tugas akhir ini.

Surabaya, Juli 2005

Rahmi Ika Noviana

UCAPAN TERIMA KASIH

Penulis tidak akan dapat menyelesaikan tugas akhir ini tanpa bantuan Allah melalui hamba-hamba-Nya. Melalui kesempatan ini penulis mengucapkan terima kasih dan *jazakumullahu bi ahsanal jaza* kepada:

1. Allah SWT, Rabb yang telah memberikan kekuatan dan kesabaran, hidayah dan pertolongan, segala karunia dan rahmat yang telah dilimpahkan-Nya. Segala puji bagi-Mu ya Allah, Maha Suci Engkau ya Rabbul Izzati.
2. Kedua orang tua, Papa dan Mama, penulis ucapkan rasa bhakti dan sayang yang mendalam, dengan doa yang tulus semoga Allah memberikan pahala yang melimpah atas segala jerih payah dan kesabaran beliau berdua.
3. Ibu Siti Rochimah dan Pak Imam yang telah memberikan bimbingan baik konsep maupun teknis, sehingga penulis bisa menemukan arah dalam pengerjaan tugas akhir ini.
4. Adik tercinta nan jauh di sana. Semoga Allah semakin menguatkan ikatan hati dan persaudaraan di antara kita
5. Teman-teman mantan PH dan DPP JMMI '00: Ik@, Ciput, Atik, Maymay, Ida, Rachie, Ani, Yetty, Yum, Leni, Rifa, serta teman-teman ikhwan. *Syukron katsiir* atas perhatian, dukungan, dan pengertiannya sehingga penulis bisa menyelesaikan tugas akhir dalam kehangatan *ukhuwah*. Semoga Allah SWT senantiasa menjaga kita dalam barisan dakwah hingga *syahid* menjemput. Aamiin

6. Teman-teman Noceng: Fetty, Lintang, Shofi, Suci, Ruli, Nindi, Yunita, Reni, Erma, Mbak Ulfa, Iin, dan teman-teman yang lain. Semoga kenangan di antara kita dapat menjadi pelajaran berharga bagi perjalanan hidup berikutnya
7. *My beloved ummi' and sisters. Sukron kabiir* atas kebersamaan dan persaudaraannya selama ini, sehingga membuat penulis selalu bisa mencintai *antum* semua karena Allah.
8. Adik-adik yang telah memberikan inspirasi dan motivasi kepada penulis untuk terus berbuat lebih baik. Jangan pernah menyerah!
9. Mbak-mbak yang telah memberikan bantuan secara teknis dan moral. Buat Mbak Nanum "*It's really a big surprise 😊*"
10. Saudara-saudaraku seperjuangan dan pihak-pihak lain yang tidak bisa disebutkan satu persatu, terima kasih semua.

ABSTRAK

Dalam pembuatan sistem aplikasi basis data berbasis web, perancangan antarmuka pengguna (presentation layer) dan lapisan bisnis (bussiness layer) merupakan tahap yang dilalui setelah pemahaman terhadap kebutuhan pengguna sistem. Adanya pola atau keteraturan dalam implementasi tahap ini menyebabkan pengembangan sistem akan lebih efisien jika menggunakan suatu aplikasi yang dapat menghasilkan kerangka dasar aplikasi web dengan cepat untuk kedua layer tersebut dan bahkan beserta kode programnya.

Untuk memenuhi kebutuhan tersebut, maka pada Tugas Akhir ini akan diimplementasikan suatu perangkat lunak yang selanjutnya disebut QuickASP. QuickASP membangkitkan kode ASP untuk membangun homepage otomatis. Untuk membangkitkan kode ASP, QuickASP membutuhkan komponen berupa basis data dan file Cascading Style Sheets (CSS). Proses awal yang dilakukan QuickASP dalam membangkitkan kode program ASP adalah membaca informasi basis data berupa tabel-tabel, nama field dan tipe data. Setelah itu QuickASP akan membangkitkan file-file ASP beserta file-file pendukungnya berdasarkan hasil pengaturan tampilan halaman web yang dilakukan oleh pengguna.

Uji coba QuickASP dilakukan pada tiga jenis basis data yaitu Microsoft Access, Microsoft SQL Server, dan Oracle. Dari hasil uji coba tersebut, QuickASP dapat membangkitkan homepage otomatis beserta fungsi-fungsi yang disediakan untuk modifikasi record dan fungsi navigasi.

DAFTAR ISI

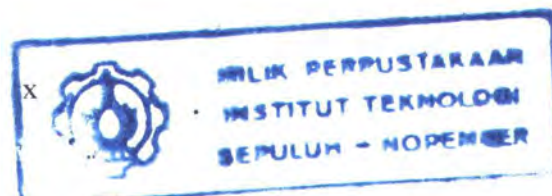
KATA PENGANTAR	iii
UCAPAN TERIMA KASIH.....	iv
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan dan Manfaat	2
1.3 Permasalahan	3
1.4 Batasan Masalah	4
1.5 Metodologi	4
1.6 Sistematika Laporan.....	7
BAB II DASAR TEORI.....	10
2.1 Active Data Object (ADO)	10
2.2 Properti Basis Data SQL Server	11
2.2.1 View Information_Schema	11
2.2.2 Columnproperty().....	12
2.3 Properti Basis Data Microsoft Access	13
2.4 Properti Basis Data Oracle.....	13
2.5 Konsep Aplikasi Berbasis Web	15
2.6 Active Server Pages	16

2.7	Interaksi Manusia dan Komputer.....	19
2.7.1	Warna	19
2.7.2	Bentuk	20
2.7.3	Tipografi.....	21
BAB III ANALISIS DAN PERANCANGAN PERANGKAT LUNAK.....		22
3.1	Deskripsi Umum Sistem	22
3.2	Pemodelan Proses	25
3.3	Perancangan Tampilan Halaman Web.....	29
3.3.1	Halaman-Halaman Hasil Pembangkitan	30
3.3.2	Tampilan Tipe Masukan dan Validasi Data.....	31
3.3.3	Sekuritas Pengunjung Situs Web	32
BAB IV IMPLEMENTASI PERANGKAT LUNAK		34
4.1	Lingkungan Implementasi	34
4.2	Implementasi Antarmuka Pengguna QuickASP.....	35
4.2.1	Bagian Form.....	35
4.2.1.1	Form Pemilihan Proyek	35
4.2.1.2	Form Pemilihan dan Pemasukan Info Basis Data.....	37
4.2.1.3	Form Pemilihan Tabel dan Metode Pembangkitan.....	40
4.2.1.4	Form Pengaturan Tampilan Halaman Daftar (1)	42
4.2.1.5	Form Pengaturan Tampilan Halaman Daftar (2)	44
4.2.1.6	Form Pengaturan Tampilan Halaman Ubah (1).....	54
4.2.1.7	Form Pengaturan Tampilan Halaman Ubah (2).....	55
4.2.1.8	Form Pengaturan Tampilan Halaman Pencarian	66

4.2.1.9	Form Pengaturan Tampilan Halaman Login.....	67
4.2.1.10	Form Pemilihan <i>Stylesheet</i> dan Direktori Keluaran.....	70
4.2.1.11	Form Pembangkitan Kode Program ASP	71
4.2.2	Bagian Tombol.....	74
4.2.3	Bagian Shortcut.....	75
5 BAB V	UJI COBA DAN EVALUASI	78
5.1	Lingkungan Uji Coba.....	78
5.1.1	Server	78
5.1.2	Client.....	79
5.2	Skenario Uji Coba.....	80
5.2.1	Skenario Pertama	80
5.2.2	Skenario Kedua	94
5.2.3	Skenario Ketiga	96
5.2.4	Skenario Keempat	100
6 BAB VI	PENUTUP	107
6.1	Kesimpulan	107
6.2	Kemungkinan Pengembangan	107
7 DAFTAR PUSTAKA.....		109
8 LAMPIRAN.....		110
a.	Tabel View Information_Schema SQL Server.....	110
b.	Tabel ColumnProperty() SQL Server	111
c.	Tabel OpenSchema() Microsoft Access	112

DAFTAR GAMBAR

Gambar 2.1	Model Obyek ADO	11
Gambar 2.2	Arsitektur Three-tier.....	16
Gambar 2.3	Ilustrasi Kombinasi Skrip ASP [6].....	17
Gambar 3.1	Gambaran Umum Sistem QuickASP	23
Gambar 3.2	DFD Level 0.....	25
Gambar 3.3	DFD Level 1.....	26
Gambar 3.4	DFD Level 2 Proses 1.1	28
Gambar 3.5	DFD Level 2 Proses 1.5	29
Gambar 4.1	Form Pemilihan Proyek (pilihan pertama).....	35
Gambar 4.2	Form Pemilihan Proyek (pilihan ke dua)	36
Gambar 4.3	Form Pemilihan dan Pemasukan Info Basis Data MS Access.....	37
Gambar 4.4	Form Pemilihan dan Pemasukan Info Basis Data SQL Server	38
Gambar 4.5	Form Pemilihan dan Pemasukan Info Basis Data Oracle	39
Gambar 4.6	Form Pemilihan Tabel dan Metode Pembangkitan.....	40
Gambar 4.7	Form Pengaturan Tampilan Halaman Daftar (1)	42
Gambar 4.8	Form Pengaturan Tampilan Halaman Daftar(2)	44
Gambar 4.9	Pengaturan Kolom <i>Label</i> pada Form Pengaturan Halaman Daftar	45
Gambar 4.10	Pengaturan Kolom <i>Detail Page</i> pada Form Pengaturan Halaman Daftar	46
Gambar 4.11	Pengaturan Kolom <i>View As</i> pada Form Pengaturan Halaman Daftar	47



Gambar 4.12	Pengaturan Kolom <i>View Format</i> pada Form Pengaturan Halaman Daftar	50
Gambar 4.13	Form Pengaturan Tampilan Tipe Data <i>Text</i>	50
Gambar 4.14	Form Pengaturan Tampilan Tipe Data <i>Currency</i>	51
Gambar 4.15	Form Pengaturan Tampilan Tipe Data <i>True/False</i>	52
Gambar 4.16	Form Pengaturan Tampilan Tipe Data <i>Image</i>	53
Gambar 4.17	Form Pengaturan Tampilan Tipe Data <i>Date</i>	53
Gambar 4.18	Form Pengaturan Tampilan Halaman Ubah (1)	54
Gambar 4.19	Form Pengaturan Tampilan Halaman Ubah (2)	55
Gambar 4.20	Form Pengaturan Jenis Masukan <i>Text Field</i>	58
Gambar 4.21	Form Pengaturan Jenis Masukan <i>Text Area</i>	60
Gambar 4.22	Form Pengaturan Jenis Masukan <i>Combobox</i> (Pilihan Pertama) ..	61
Gambar 4.23	Form Pengaturan Jenis Masukan <i>Combobox</i> (Pilihan Kedua) ..	62
Gambar 4.24	Form Pengaturan Jenis Masukan <i>File-based Image</i> dan <i>Database Image</i>	63
Gambar 4.25	Form Pengaturan Jenis Masukan <i>Date</i>	64
Gambar 4.26	<i>Date Time Picker</i>	65
Gambar 4.27	Form Pengaturan Tampilan Halaman Pencarian	66
Gambar 4.28	Form Pengaturan Tampilan Halaman Login	67
Gambar 4.29	Form Pengaturan Tampilan Halaman Login (Tingkat Lanjut) ..	68
Gambar 4.30	Form Pemilihan <i>Stylesheet</i> dan Direktori Keluaran	70
Gambar 4.31	Form Pembangunan Kode ASP	71
Gambar 4.32	Macam-macam Tombol pada QuickASP	74

Gambar 4.33	<i>Shortcut</i> pada QuickASP.....	76
Gambar 5.1	Struktur Basis Data <i>Northwind</i> pada MS SQL Server.....	81
Gambar 5.2	Form Pemilihan Proyek (Skenario 1)	82
Gambar 5.3	Form Pemilihan Basis Data	82
Gambar 5.4	Form Pemilihan Tabel dan Metode Pembangkitan (Skenario 1).....	83
Gambar 5.5	Form Pemilihan <i>Field</i> Halaman Daftar (Skenario 1).....	84
Gambar 5.6	Form Pengaturan Halaman Daftar (<i>default</i>)	85
Gambar 5.7	Form Pengaturan Halaman Daftar (setelah diubah).....	85
Gambar 5.8	Form Pengaturan Susunan <i>Field</i> pada Halaman Ubah (Skenario 1).....	86
Gambar 5.9	Form Pengaturan Halaman Ubah Skenario 1 (<i>default</i>).....	87
Gambar 5.10	Desain Basis Data “Northwind” Tabel “Employees”	88
Gambar 5.11	Form Pengaturan Halaman Ubah Skenario 1 (setelah diubah) .	88
Gambar 5.12	Form Pengaturan Halaman Pencarian (Skenario 1).....	89
Gambar 5.13	Form Pengaturan Halaman Login (Skenario 1)	90
Gambar 5.14	Pemilihan <i>Stylesheet</i> (Skenario 1).....	90
Gambar 5.15	Pemilihan Direktori Keluaran dan URL Website (Skenario 1).	91
Gambar 5.16	Form Pembangkitan Kode Program (Skenario 1).....	91
Gambar 5.17	Tampilan Halaman Daftar (Skenario 1).....	92
Gambar 5.18	Tampilan Halaman Ubah (Skenario 1)	92
Gambar 5.19	Tampilan Halaman Tampil (Skenario 1).....	93
Gambar 5.20	Tampilan Halaman Tambah (Skenario 1).....	94

Gambar 5.21	Form Pemilihan Tabel dan Metode Pembangkitan (Skenario 2).....	95
Gambar 5.22	File-File Hasil Pembangkitan Kode pada Skenario Kedua.....	96
Gambar 5.23	Form Pemilihan Proyek yang Sudah Ada (Skenario 3)	97
Gambar 5.24	Form Pengaturan Halaman Daftar pada Proyek yang Dibuka (Skenario 3).....	98
Gambar 5.25	Form Pengaturan Halaman Login (Skenario 3)	99
Gambar 5.26	Tampilan Halaman Login (Skenario 3).....	99
Gambar 5.27	Properti Kolom dari Tabel “ANGGOTA” (Skenario 4)	100
Gambar 5.28	Pengaturan Format Tampilan Halaman Ubah (Skenario 4)....	101
Gambar 5.29	Pengaturan Format Tampilan Halaman Login (Skenario 4)...	102
Gambar 5.30	Pengaturan Format Tampilan Halaman Login Lanjut (Skenario 4).....	103
Gambar 5.31	Tampilan Halaman Login	104
Gambar 5.32	Tampilan Halaman Daftar.....	104
Gambar 5.33	Pesan Error	105
Gambar 5.34	Tampilan Halaman Ubah	105
Gambar 5.35	Login sebagai Tamu.....	106
Gambar 5.36	Halaman Daftar untuk Login sebagai Tamu	106

DAFTAR TABEL

Tabel 2.1	Arti universal dari bentuk secara psikologi.....	20
Tabel 3.1	Tipe Masukan Berdasarkan Tipe Data.....	31
Tabel 3.2	Jenis Validasi Data.....	32
Tabel 4.1	Jenis Tampilan pada Halaman Daftar dan Halaman Detail Berdasarkan Tipe Data.....	47
Tabel 4.2	Jenis Masukan pada Halaman Ubah dan Halaman Tambah Berdasarkan Tipe Data.....	57
Tabel 4.3	Jenis Validasi Data pada Pengaturan Tampilan Halaman Ubah dan Halaman Tambah	59

BAB I

PENDAHULUAN

Bab ini berisi hal-hal yang berkaitan dengan tugas akhir secara umum meliputi latar belakang pembuatan tugas akhir, tujuan serta manfaat tugas akhir yang dibuat, permasalahan yang akan dihadapi, metodologi pembuatan tugas akhir, dan sistematika laporan tugas akhir.

1.1 Latar Belakang

Seiring dengan semakin pesatnya perkembangan internet, teknologi web dari waktu ke waktu juga mengalami perkembangan yang pesat. Saat ini web bukan lagi merupakan sebuah halaman on-line yang hanya bisa dibaca atau statis, namun telah berkembang menjadi web yang lebih dinamis dan interaktif, yang mampu menerima dan memberikan respon dari dan ke pengakses. Selain itu sejalan dengan perkembangan teknologi web, nilai manfaat dari keberadaan web bagi masyarakat juga mengalami peningkatan. Pada mulanya web hanya berfungsi sebagai penyedia layanan informasi secara on-line. Namun saat ini web telah berkembang sebagai salah satu aspek penunjang dalam dunia bisnis yang diwujudkan dalam bentuk *e-commerce*, *e-marketing*, *on-line transaction*, dan lain-lain.

Salah satu bahasa pemrograman yang dapat dipakai untuk membuat sebuah web yang dinamis adalah ASP (*Active Server Pages*). ASP adalah suatu skrip yang bersifat *server-side* yang memiliki kemampuan untuk

dikombinasikan dengan teks, HTML, dan komponen-komponen lain untuk membuat halaman web yang lebih menarik, dinamis, dan interaktif.

Pada umumnya seorang pengembang web harus memahami sintaks ASP dan HTML serta bahasa SQL jika hendak mengembangkan sebuah situs web dengan bahasa pemrograman ASP. Padahal tidak semua orang mempunyai cukup waktu untuk mempelajari lebih dalam tentang pemrograman web (ASP) dan pemrograman basis data (SQL) di depan komputer. Selain itu jika pengembang ingin melakukan desain ulang pada basis data yang digunakan, misal ingin mengganti tipe data sebuah kolom pada sebuah tabel basis data, pengembang web juga harus mengubah desain pada basis data. Akibatnya, kode program ASP yang telah dibuat pun harus diubah. Hal ini semakin membuat terasa sulit jika ingin membuat sebuah web yang dinamis.

Untuk itu, tugas akhir ini akan membuat sebuah perangkat lunak agar pembuatan web dinamis berbasis ASP menjadi lebih mudah dan cepat. Perangkat lunak yang dibuat secara otomatis akan menghasilkan kode program ASP kemudian mengirim data ke URL yang diinginkan.

1.2 Tujuan dan Manfaat

Tujuan utama dari pembuatan tugas akhir ini adalah untuk membuat perangkat lunak yang mampu membangkitkan kode program ASP berdasarkan pembacaan struktur basis data asal. Perangkat lunak juga

dilengkapi fitur untuk mempermudah pengguna melakukan modifikasi pada desain tampilan web sesuai kehendaknya.

1.3 Permasalahan

Permasalahan yang diangkat dalam Tugas Akhir ini adalah :

1. Bagaimana agar perangkat lunak yang dibuat mampu membaca struktur basis data yang dimasukkan pengguna. Misal nama-nama tabel, nama-nama kolom untuk tiap tabel beserta tipe data dan panjangnya, serta relasi antar tabel.
2. Bagaimana agar perangkat lunak yang dibuat menyediakan fasilitas yang memungkinkan pengguna untuk melakukan modifikasi desain antar muka pada situs web meliputi desain tampilan secara keseluruhan, tipe masukan yang digunakan (*button, textbox, checkbox, combobox, listbox, radio button*), serta navigasi *record*.
3. Bagaimana membuat perangkat lunak yang dapat berfungsi sebagai pembangkit kode program ASP untuk pembuatan web yang dinamis dengan berbagai pilihan *provider* basis data
4. Bagaimana agar perangkat lunak yang dibuat mampu menyesuaikan kode program ASP hasil jika dilakukan perubahan pada desain basis data asal dengan melakukan pembangkitan ulang.

1.4 Batasan Masalah

Dari permasalahan-permasalahan di atas, maka batasan dalam tugas akhir ini adalah :

1. Basis data sumber yang dapat dibaca oleh perangkat lunak ini dibatasi pada basis data Microsoft Access, SQL Server, dan Oracle.
2. Perangkat lunak mampu melakukan pembaruan pada kode program ASP jika dilakukan perubahan struktur *field* pada tabel basis data sumber dengan tipe data yaitu *text*, *numeric*, *date*, *boolean* dan *image* (.gif, .jpg, .bmp).
3. Perangkat lunak yang dibuat merupakan aplikasi *desktop*.

1.5 Metodologi

Metodologi yang digunakan dalam pembuatan tugas akhir ini adalah sebagai berikut :

1. Studi Literatur dan Pemahaman Sistem

Tahap ini merupakan tahap pengumpulan informasi yang berhubungan dengan konsep-konsep dan teknik tentang bahasa pemrograman ASP, Visual Basic 6, serta teknik pemrograman basis data yang meliputi Oracle 9i, SQL Server dan Microsoft Access. Konsep tersebut diperoleh dari berbagai referensi (buku teks, *paper*, maupun artikel-artikel di internet).

2. Perancangan Sistem

- Perancangan Data

Pada tahap ini akan dirancang data masukan dan data keluaran yang dihasilkan perangkat lunak.

- Perancangan Proses

Pada tahap ini akan dirancang proses-proses yang dilakukan oleh perangkat lunak mulai pemilihan jenis basis data, pemilihan tabel, proses pengaturan format halaman web untuk tiap tabel, sampai proses pembangkitan kode program ASP

- Perancangan Antarmuka Pengguna

Antarmuka pengguna yang bersifat *user friendly* dirancang untuk memudahkan pengguna dalam menggunakan aplikasi QuickASP ini.

3. Implementasi Sistem dan Pembuatan Perangkat Lunak

Berdasarkan rancangan yang telah dibuat pada tahap sebelumnya, pada tahap ini dilakukan implementasi sistem serta pembuatan perangkat lunak. Perangkat lunak dibuat dengan menggunakan Microsoft Visual Basic 6.

- a. Pengambilan properti basis data yang meliputi nama tabel, nama view, nama kolom, tipe data, panjang kolom, *primary key*, *foreign key*.
- b. Pembangkitan file proyek dengan ekstensi *.qap (Quick ASP Project)* yang berisi data desain halaman web sesuai pengaturan yg dilakukan pengguna

- c. Pembangkitan kode program ASP beserta file-file pendukung yang diperlukan (JavaScript, CSS)

4. Uji Coba dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap sistem perangkat lunak yang telah selesai dikembangkan melalui beberapa skenario yaitu:

- Melakukan pembangkitan kode program ASP dengan cara membuat proyek baru maupun membuka proyek yang sudah ada
- Melakukan pembangkitan kode program ASP dari berbagai basis data (Oracle, SQL Server, dan Microsoft Access)
- Melakukan pembangkitan kode program ASP baik secara manual (dengan melakukan pengaturan tiap halaman web yang akan dihasilkan) maupun secara otomatis (menggunakan format halaman web yang telah disiapkan oleh aplikasi)
- Melakukan pengaturan tampilan antarmuka pengguna tiap halaman web yang meliputi *List Page*, *Detail Page*, *Add Page*, *Edit Page*, *Search Page*, dan *Login Page*
- Melakukan pembangkitan kode program ASP dengan menggunakan berbagai file CSS (*Cascading Style Sheet*) yang telah disediakan aplikasi

Setelah dilakukan uji coba terhadap perangkat lunak, kemudian dilakukan evaluasi apakah tujuan yang telah ditentukan tercapai atau tidak dengan parameter sebagai berikut:

- c. Pembangkitan kode program ASP beserta file-file pendukung yang diperlukan (JavaScript, CSS)

4. Uji Coba dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap sistem perangkat lunak yang telah selesai dikembangkan melalui beberapa skenario yaitu:

- Melakukan pembangkitan kode program ASP dengan cara membuat proyek baru maupun membuka proyek yang sudah ada
- Melakukan pembangkitan kode program ASP dari berbagai basis data (Oracle, SQL Server, dan Microsoft Access)
- Melakukan pembangkitan kode program ASP baik secara manual (dengan melakukan pengaturan tiap halaman web yang akan dihasilkan) maupun secara otomatis (menggunakan format halaman web yang telah disiapkan oleh aplikasi)
- Melakukan pengaturan tampilan antarmuka pengguna tiap halaman web yang meliputi *List Page*, *Detail Page*, *Add Page*, *Edit Page*, *Search Page*, dan *Login Page*
- Melakukan pembangkitan kode program ASP dengan menggunakan berbagai file CSS (*Cascading Style Sheet*) yang telah disediakan aplikasi

Setelah dilakukan uji coba terhadap perangkat lunak, kemudian dilakukan evaluasi apakah tujuan yang telah ditentukan tercapai atau tidak dengan parameter sebagai berikut:

- a. Kemampuan dalam membangkitkan kode program berdasarkan data masukan yaitu properti basis data, data hasil pengaturan untuk tiap halaman web, dan file CSS
- b. Konsistensi dan kestabilan sistem *homepage* yang dihasilkan dari aplikasi ini

Selanjutnya jika ditemukan kesalahan-kesalahan pada sistem yang telah dibuat maka dilakukan perbaikan-perbaikan bagi kesempurnaan sistem tersebut.

5. Penyusunan Buku Tugas Akhir

Tahap penulisan buku tugas akhir bertujuan untuk menuliskan informasi-informasi mengenai perangkat lunak yang telah dibangun. Tahap ini sekaligus merupakan tahap terakhir dari pembentukan sistem secara keseluruhan.

6. Revisi Buku Tugas Akhir

Dalam tahap ini dilakukan perbaikan-perbaikan atau penambahan-penambahan yang diperlukan terhadap buku tugas akhir yang telah dibuat.

1.6 Sistematika Laporan

Pembahasan yang akan disajikan dalam tugas akhir ini dibagi menjadi dalam beberapa bab sebagai berikut:

Bab I Pendahuluan

Berisi latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat serta metodologi dan sistematika pembahasan dari tugas akhir.

Bab II Dasar Teori

Pada bab ini akan dibahas dasar-dasar ilmu yang mendukung pengerjaan tugas akhir ini, meliputi Microsoft Visual Basic 6, SQL Server Schema, Microsoft Access MySys Object, Oracle 9i, ActiveX Data Object (ADO), Active Server Page (ASP), serta Interaksi Manusia dan Komputer (IMK).

Bab III Analisis dan Perancangan Perangkat Lunak

Perancangan sistem secara umum beserta komponen-komponen yang terlibat di dalamnya, meliputi struktur data aplikasi, pengumpulan basis data, perancangan proses kebutuhan sistem, serta perancangan antarmuka pengguna.

Bab IV Implementasi Perangkat Lunak

Pada tahap ini dilakukan implementasi pembuatan perangkat lunak berdasarkan rancangan yang telah dibuat pada tahap sebelumnya.

Bab V Uji Coba dan Evaluasi

Pada tahap ini perangkat lunak yang telah dibuat akan diuji, dievaluasi serta dilakukan penyempurnaan jika masih terdapat kekurangan. Uji coba yang dilakukan diantaranya dengan melakukan proses pembangkitan kode ASP melalui berbagai pilihan basis data, melakukan pengaturan halaman web yang hendak dibangkitkan, serta dengan memilih file CSS yang berlainan

Bab VI Penutup

Penarikan kesimpulan dan saran pengembangan lebih lanjut dari sistem yang dibuat.

BAB II

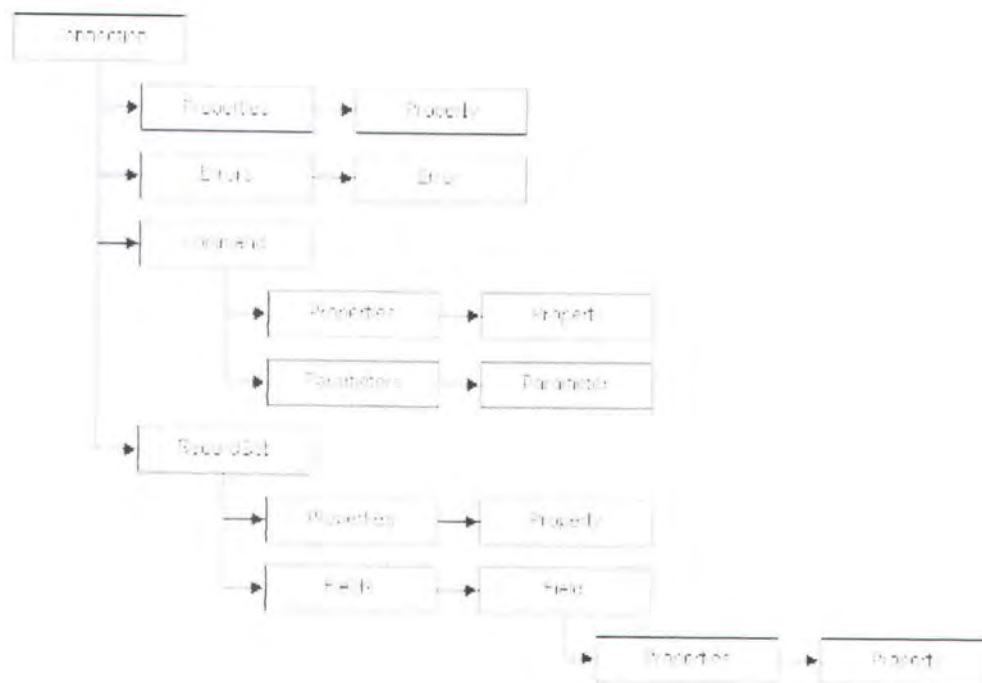
DASAR TEORI

Pada bagian ini akan dibahas konsep dasar serta teknologi yang digunakan untuk mengimplementasikan pembuatan perangkat lunak QuickASP.

2.1 Active Data Object (ADO)

ADO memiliki kemampuan untuk mengakses beragam sumber data, termasuk di dalamnya adalah basis data dalam format Microsoft Access dan Microsoft SQL Server, maupun basis data yang bukan dari Microsoft seperti basis data Oracle dan Sybase. Dan yang lebih penting lagi yaitu bahwa ADO pada akhirnya akan menggantikan DAO (Data Access Objects) dan RDO (Remote Data Objects), yang merupakan teknologi pengaksesan data cara lama. Microsoft meyakinkan bahwa ADO akan terus dikembangkan dan mendukung kompatibilitas dengan sistem operasi dan basis data masa depan.

ADO memiliki model obyek yang dapat dilihat pada gambar 2.1.



Gambar 2.1 Model Obyek ADO

Seperti dapat dilihat pada gambar 2.1, ADO memiliki model obyek yang relatif sederhana dan terdiri atas 3 (tiga) obyek utama dimana tiap-tiap obyek memiliki sejumlah properti dan metode.

2.2 Properti Basis Data SQL Server

2.2.1 View Information_Schema

Pada SQL Server versi 7.0 ke atas, informasi mengenai struktur basis data SQL Server seperti nama basis data, nama-nama tabel, struktur tabel, nama-nama view, nama-nama kolom untuk tiap tabel, tipe data masing-masing kolom, dan informasi lainnya, tersimpan dalam sebuah view yang bernama INFORMATION_SCHEMA. Masing-masing basis data memiliki view ini.

Berikut beberapa sintaks SQL yang digunakan untuk menampilkan beberapa properti basis data:

1. Untuk menampilkan daftar nama-nama basis data dimana *pengguna* tertentu memiliki hak akses tertentu:

```
select * from information_schema.schemata
```

2. Untuk menampilkan daftar nama-nama tabel

```
select * from information_schema.tables
where table_type = 'base table'
```

3. Untuk menampilkan daftar nama kolom yang merupakan *primary key*

```
select * from Information_Schema.Key_Column_Usage
where TABLE_NAME=' ' & MyTable & ' '
and constraint_name like '%PK%'
```

Daftar tabel selengkapnya yang menampilkan properti dari *view information_schema* dapat dilihat pada halaman lampiran.

2.2.2 Columnproperty()

Selain dengan cara mengakses view INFORMATION_SCHEMA, SQL Server juga menyediakan sebuah fungsi yang dapat digunakan untuk mendapatkan *basis data properties* yang lain. Fungsi tersebut adalah COLUMNPROPERTY yang mengembalikan informasi mengenai parameter dari suatu kolom atau prosedur.

Sintaks : COLUMNPROPERTY (*id* , *column* , *property*)

Argumen

id : ID (*identifier*) dari tabel atau prosedur

column : nama kolom atau parameter

property : properti yang ingin diambil nilainya

Daftar tabel selengkapnya yang menampilkan nilai serta nilai balik dari fungsi *columnproperty()* dapat dilihat pada halaman lampiran.

2.3 Properti Basis Data Microsoft Access

Informasi mengenai struktur basis data pada Microsoft Access dapat diambil dengan menggunakan fungsi *OpenSchema()* pada Microsoft Visual Basic 6.

Daftar beberapa parameter pada fungsi *OpenSchema()* beserta nama kolom yang menyertainya dapat dilihat pada halaman lampiran.

2.4 Properti Basis Data Oracle

Informasi struktur basis data Oracle tersimpan dalam beberapa tabel. Berikut beberapa sintaks SQL yang digunakan untuk menampilkan beberapa properti basis data Oracle:

1. Untuk menampilkan daftar nama-nama tabel

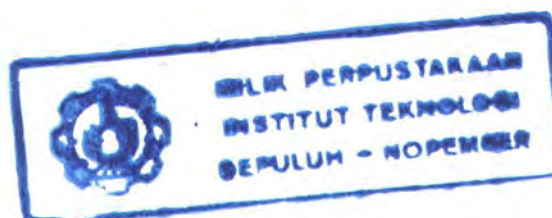
```
select TABLE_NAME from USER_TABLES
```

2. Untuk menampilkan daftar nama-nama view

```
select VIEW_NAME from USER_VIEWS
```

3. Untuk menampilkan daftar nama-nama *field* dari sebuah tabel

```
select COLUMN_NAME from USER_TAB_COLUMNS
```




```
where table_name='table_name'
```

4. Untuk menampilkan daftar properti dari sebuah *field*

```
select
    DATA_DEFAULT, DATA_TYPE, DATA_LENGTH, NULLABLE
from USER_TAB_COLUMNS
where TABLE_NAME='table_name'
and COLUMN_NAME='column_name'
```

5. Untuk menampilkan daftar *field* yang merupakan *primary key* dari sebuah tabel

```
select COLS.TABLE_NAME, COLS.COLUMN_NAME,
    COLS.POSITION, CONS.STATUS, CONS.OWNER
from ALL_CONSTRAINTS CONS, ALL_CONS_COLUMNS COLS
where COLS.TABLE_NAME = 'table_name'
and CONS.CONSTRAINT_TYPE = 'P'
and CONS.CONSTRAINT_NAME = COLS.CONSTRAINT_NAME
and CONS.OWNER = COLS.OWNER
order by COLS.TABLE_NAME, COLS.POSITION
```

6. Untuk menampilkan daftar *field* yang merupakan *Foreign Key* dari sebuah tabel

```
select a.CONSTRAINT_NAME, a.CONSTRAINT_TYPE,
    b.TABLE_NAME MYTABLE, b.COLUMN_NAME MYCOLUMN
from ALL_CONSTRAINTS A, ALL_CONS_COLUMNS B
where a.OWNER='OWNER_NAME'
and a.CONSTRAINT_TYPE='R'
and b.CONSTRAINT_NAME=a.R_CONSTRAINT_NAME
```

2.5 Konsep Aplikasi Berbasis Web

Dalam tugas akhir ini, perangkat lunak akan membangkitkan kode asp sebagai hasil dari proses pembangkitan kode. Pada sub-bab ini akan dijelaskan konsep aplikasi berbasis web. Aplikasi berbasis web memiliki bagian dasar atau lapisan sebagai berikut :

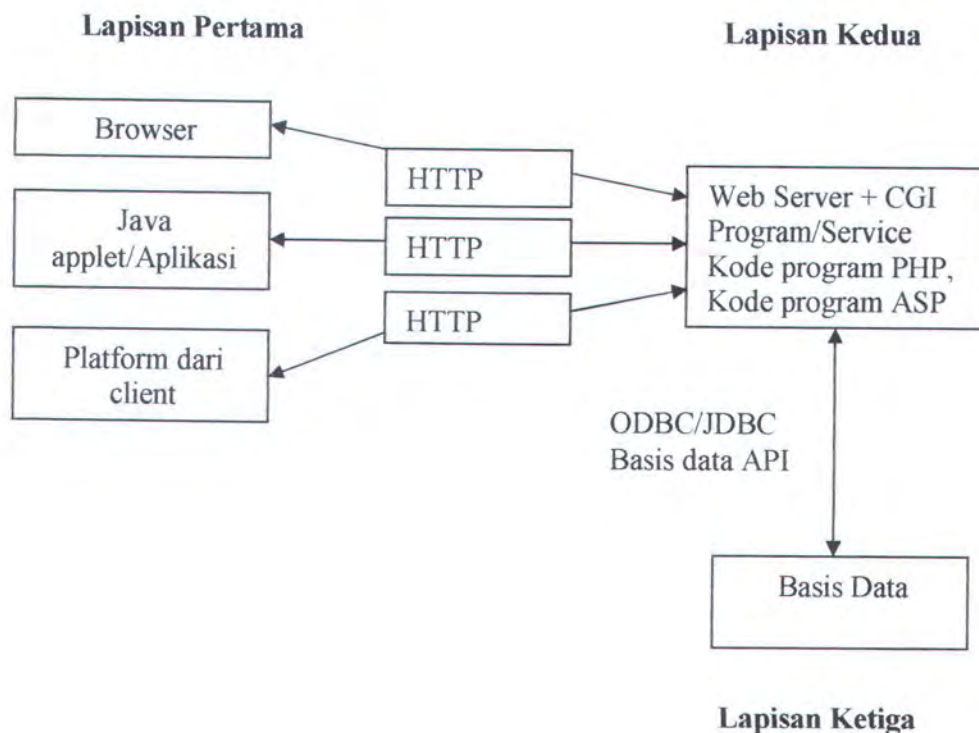
- *Client*, yang termasuk di dalamnya antara lain : *web browser* dari pengguna, *java applet*, aplikasi *java*, atau platform dari program *client* yang berdiri sendiri.
- Aplikasi logika, meliputi : algoritma pengkodean pada kode program CGI, modul khusus dari *web server*, atau aplikasi *server* yang berdiri sendiri.
- Konektifitas basis data, meliputi : API dari basis data, protokol konektifitas umum seperti ODBC atau JDBC.
- *Server* untuk basis data, seperti : RDBMS, ODBMS dan lain-lain.[3].

Implementasi dari beberapa aplikasi dapat menggunakan model *multitier*, karena satu atau beberapa lapisan dapat dipakai bersama-sama. Namun implementasi secara umum adalah sistem *three-tier* (tiga lapisan), yang terdiri dari tiga komponen utama, yaitu :

- Lapisan pertama, yang merupakan aplikasi dari *client*, contohnya : *browser* dari pengguna
- Lapisan kedua, merupakan aplikasi *web server*, Kode program CGI dan API koneksi untuk basis data, seperti *Apache web server* dengan modul PHPnya, IIS ASP.

- Lapisan ketiga adalah *server* untuk basis data.[2].

Tiga lapisan tersebut dapat digambarkan sebagai berikut :



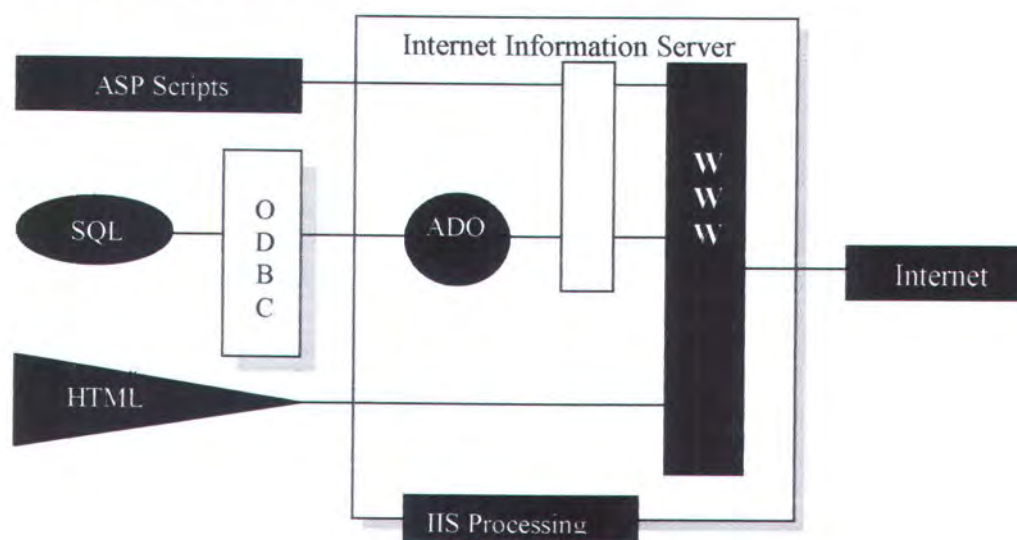
Gambar 2.2 Arsitektur Three-tier

2.6 Active Server Pages

Pada dasarnya bahasa-bahasa yang dapat digunakan untuk membangun suatu web dapat dibedakan menjadi dua, yaitu *Client-side* dan *Server-side*. *Client-side* memiliki arti bahwa informasi yang disampaikan akan dieksekusi di *client* atau *browser*. Adapun lawannya adalah *Server-side*, yaitu proses pengerjaan informasi akan dikirim dan diproses di server web tersebut.

Active Server Pages (ASP) adalah suatu skrip yang bersifat *server-side* yang memiliki kemampuan untuk dikombinasikan dengan teks, HTML, dan komponen-komponen lain untuk membuat halaman web yang lebih menarik, dinamis, dan interaktif. Komponen adalah objek yang sudah terkompilasi dengan *native code* di masing-masing platform, baik itu platform Windows maupun platform lainnya. Dalam ASP dapat digunakan sintaks pemrograman VBCode program ataupun Jkode program sehingga *web programmer* dapat secara bebas memilih salah satu bahasa pemrograman yang dikuasai atau disenanginya.

Dalam pembuatan aplikasi yang interaktif antara pemilik web dan pengunjung dengan menggunakan ASP diperlukan dukungan bahasa lain. Sebagai contoh, apabila ingin membuat aplikasi yang melibatkan penggunaan basis data maka perlu juga digunakan bahasa SQL.



Gambar 2.3 Ilustrasi Kombinasi Skrip ASP [6]

ASP dimaksudkan untuk menggantikan teknologi lama yang bersifat *server-side*, seperti CGI (Common Gateway Interface) yang memiliki beberapa kelemahan dan berjalan di lingkungan UNIX. Ciri yang dimiliki oleh aplikasi yang menggunakan ASP salah satunya adalah bahwa aplikasi tersebut memiliki ekstensi **.asp**.

Kelebihan-kelebihan ASP dalam pembuatan aplikasi dinamis dibandingkan dengan yang lain yaitu:

- Sintaks-sintaksnya mudah dipelajari karena tidak mengenal pendeklarasian variabel dan akses tingkat rendah lainnya
- Kode program terintegrasi dengan *file* HTML sehingga memudahkan pembuatan desain tampilan dan dapat berjalan lebih cepat dibandingkan dengan CGI
- ASP merupakan kode program yang berorientasi pada objek dan dapat dikembangkan lebih jauh dengan menggunakan komponen-komponen *ActiveX server* atau ADO
- Tidak ada proses *compiling* atau *linking*
- ASP memiliki fleksibilitas, dapat berinteraksi dengan aplikasi-aplikasi web yang dibuat dengan bahasa CGI, ISAP, dan kode program lainnya

Untuk mencoba, kode program ASP dapat dijalankan pada sebuah PC berbasis Windows tanpa terhubung ke internet dengan terlebih dahulu menginstall *Personal Web Server (PWS)* atau *Internet Information Server (IIS)*.

Pemrograman ASP merupakan program kode program sebagai alat bantu pembuatan *homepage*. Halaman-halaman web akan bergerak secara dinamis

dengan adanya program kode program, sedangkan form-form HTML cenderung statis. Program ASP selalu ditandai blok program `<%` dan akhir blok `%>`. Tanda tersebut bisa disisipkan di sembarang tempat kode HTML dan file ASP harus disimpan dengan akhiran `*.asp`, sedangkan tempat file harus di direktori yang dieksekusi oleh web server. *Default* tempat biasanya adalah `c:\webshare` atau `c:\inetpub\wwwroot`.

2.7 Interaksi Manusia dan Komputer

Dalam tugas akhir ini, perangkat lunak akan membangkitkan kode ASP yang digunakan untuk membangun sebuah *homepage*. Pada sub bab ini akan dijelaskan konsep interaksi manusia dan komputer, terutama konsep yang berhubungan dengan perangkat lunak dan *homepage* yang dibangun.

2.7.1 Warna

Pemilihan warna adalah satu hal yang sangat penting dalam menentukan respon dari pengunjung. Warna adalah hal yang pertama dilihat oleh seorang pengunjung (terutama warna latar belakang), dan dalam pembuatan situs bisa membuat situs untuk menampilkan warna dulu sementara isi yang lain (tulisan dan gambar) masih dalam proses download. Cara ini akan membuat kesan untuk seluruh situs itu.

Untuk mencapai desain warna yang efektif, bisa dimulai dengan memilih warna yang bisa merepresentasikan tujuan dari situs. *Pallet* warna yang dipakai sebaiknya cocok dengan pribadi dan tujuan dari situs tersebut. Jika misalnya

situsnya adalah untuk situs komunitas, maka sebaiknya memilih warna-warna hangat untuk membuat suasana lebih santai. Jika situsnya adalah untuk situs informasi, dimana isi akan mendominasi, maka warna sebaiknya sederhana dan tidak mengganggu (misalnya jangan menggunakan latar belakang dengan warna mencolok).

2.7.2 Bentuk

Bentuk juga bisa digunakan untuk menarik respon dari pengunjung. Penggunaan bentuk yang efektif secara psikologi akan memotivasi pengunjung, menginspirasi pengunjung dan memberikan tantangan kepada pengunjung.

Tabel 2.1 Arti universal dari bentuk secara psikologi

Bentuk	Respon Psikologis
Lingkaran	<ul style="list-style-type: none"> - Koneksi, komunitas, keseluruhan, ketahanan, pergerakan, keamanan. - Referensi untuk perasaan kewanitaan : kehangatan, kenyamanan, sensualitas, dan cinta.
Kotak	<ul style="list-style-type: none"> - Keteraturan, logis, keamanan. - Kotak juga adalah dasar dari objek 3 dimensi yang berarti berat, massa, dan kepadatan.
Segitiga	<ul style="list-style-type: none"> - Energi, power, keseimbangan, hukum, ilmu pasti, agama. - Juga sebagai referensi untuk perasaan maskulin : kekuatan, agresi, dan pergerakan yang dinamik.

Bentuk biasanya dapat digabungkan untuk membuat kesan yang lebih kuat. Misalnya penggunaan sebuah lingkaran dan sebuah segi tiga, akan menghasilkan kesan berenergi, dan dinamik. Atau penggunaan sebuah lingkaran dan sebuah persegi untuk kehangatan dan perasaan aman.

Kekuatan dari bentuk dan kombinasi bentuk telah ditujukan di berbagai bidang, seperti automotif, bangunan, kemasan produk, dan logo perusahaan.

2.7.3 Tipografi

Tipografi adalah sebuah seni cukup rumit, apalagi kerumitannya ditambah dengan medium desain web yang terbatas. Huruf di web dapat dibuat sebagai bagian dari gambar, atau dengan cara HTML, atau dengan *stylesheets*.

Dengan cara HTML dan *stylesheets*, kadang akan didapatkan hasil yang kurang memuaskan atau kurang bagus karena bentuk huruf yang digunakan mungkin tidak akan tersedia di komputer si pengunjung. Jika komputer pengunjung tidak mempunyai bentuk huruf yang telah ditentukan oleh pembuat web, maka yang akan digunakan adalah tampilan menggunakan bentuk huruf alternatif atau bentuk huruf standard.

Perancang dapat menentukan pilihan tentang huruf yang mereka buat dengan gambar, dan juga bentuk huruf umum yang dipakai pada isi tulisan. Begitu banyak bentuk huruf, sampai sulit untuk mengatakan berapa banyak bentuk huruf yang ada. Sehingga agak sulit untuk menentukan efek atau pengaruh yang bisa ditimbulkan kepada pengunjung.

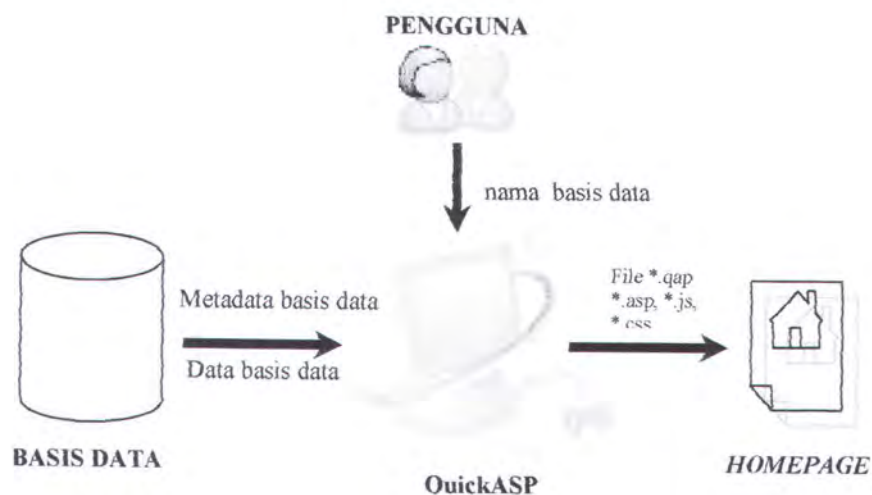
BAB III

ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak yang meliputi deskripsi umum sistem, pemodelan proses dengan *Data Flow Diagram*, serta perancangan tampilan halaman web yang dikehendaki sebagai bagian dari proses perancangan sistem secara keseluruhan. Pemodelan data yang meliputi CDM (*Conceptual Data Modelling*) dan PDM (*Phisically Data Modelling*) tidak dibuat karena aplikasi QuickASP ini tidak membutuhkan basis data sebagai penyimpan proyek maupun kode program yang dibangkitkan.

3.1 Deskripsi Umum Sistem

Secara garis besar, perangkat lunak yang hendak dibuat berfungsi sebagai pembangkit kode program ASP beserta kode program lain sebagai pendukung untuk menghasilkan halaman web dinamis. Disebut dinamis karena pengunjung web dapat melakukan proses penambahan, pengubahan serta penghapusan data suatu tabel pada basis data tertentu. Dengan demikian, perlu dilakukan pembukaan dan pembacaan isi maupun struktur basis data.



Gambar 3.1 Gambaran Umum Sistem QuickASP

Dari gambar di atas terlihat bahwa pengguna cukup memasukkan nama basis data yang hendak dibangkitkan ke dalam kode program ASP serta memilih *stylesheet* yang telah disediakan oleh aplikasi. Selanjutnya aplikasi akan membaca isi dan struktur basis data tersebut guna pengaturan tampilan masing-masing halaman web. Kemudian pada akhirnya aplikasi akan membangkitkan kode program ASP dan beberapa kode program pendukung lainnya.

Secara detail proses-proses yang dilakukan pengguna yaitu:

- Memilih antara membuat proyek baru atau membuka proyek yang sudah ada
- Memilih direktori untuk menyimpan file proyek yang berekstensi *.qap. File yang berekstensi *.qap (*QuickASP Project*) ini terbentuk setiap saat aplikasi QuickASP selesai melakukan pengaturan halaman web yang akan dihasilkan. File ini berisi kode-kode hasil pengaturan halaman web yang diperlukan oleh QuickASP guna pembangkitan kode ASP.

- c. Memasukkan nama basis data beserta propertinya jika diperlukan (*username* dan *password*)
- d. Memilih tabel yang hendak dibangkitkan ke dalam kode ASP
- e. Memilih antara melakukan pembangkitan kode program secara manual (tahap demi tahap) atau melakukan pembangkitan kode program secara otomatis oleh perangkat lunak
- f. Melakukan modifikasi format tampilan masing-masing halaman web sesuai keinginan pengguna yang meliputi halaman daftar (*List Page*), halaman pengubahan data (*Edit Page*), halaman pencarian data (*Search Page*), dan halaman login (*Login Page*). Proses ini bersifat opsional karena telah disediakan format tampilan *default* untuk masing-masing halaman web.
- g. Memilih *stylesheet* (*.css) yang telah disediakan untuk mengatur tampilan *homepage* ketika dilihat dari *web browser*
- h. Menentukan direktori tempat penyimpanan file-file hasil pembangkitan yang berisi kode ASP serta kode program pendukung lainnya (*Java Script*)
- i. Menentukan URL untuk menampilkan halaman *default* (*default page*) dalam *web browser*

File-file yang dibangkitkan merupakan kode ASP untuk membangun *homepage*. Komponen basis data pada sistem *homepage* merupakan sumber data bagi *homepage* tersebut. Pada saat melakukan proses pembangkitan kode ASP, basis data akan memberikan informasi berupa struktur tabel-tabel, nama *field* dan tipe *field* yang ada didalamnya. Struktur tabel akan digunakan untuk menentukan

dilakukan pembacaan struktur basis data meliputi nama tabel, nama *field*, tipe data, panjang maksimum karakter, *primary key*, dan *foreign key*.

b. Proses Pengaturan Format Tampilan Halaman Web

Setelah dilakukan pembacaan struktur basis data, kemudian aplikasi menampilkan format *default* tampilan untuk tiap-tiap halaman web. Format tampilan ini meliputi navigasi *record* pada halaman daftar (*List Page*), tipe masukan pada halaman pengubahan data (*Edit Page*) yang telah disesuaikan dengan tipe data masing-masing *field*, tampilan halaman pencarian (*Search Page*), serta tampilan halaman login (*Login Page*).

c. Proses Pemilihan Direktori Keluaran

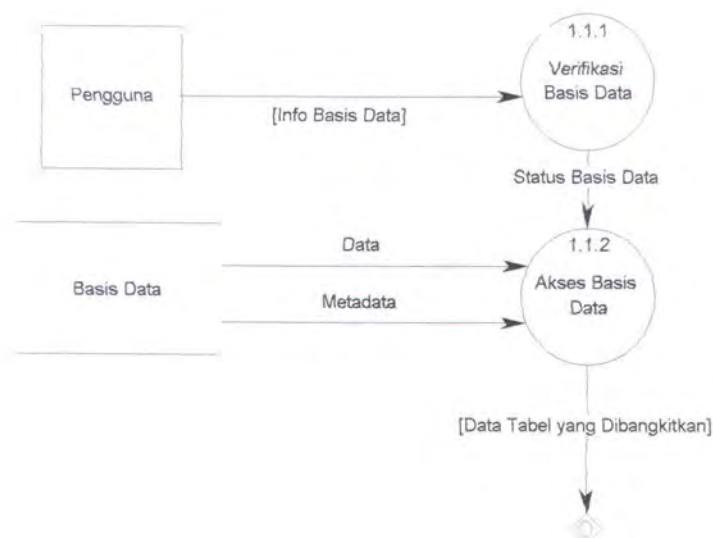
Proses ini digunakan untuk memilih sebuah direktori di *server* yang akan digunakan sebagai tempat menyimpan file-file kode ASP yang dibangkitkan.

d. Proses Pembangkitan File Proyek QuickASP

Proses ini dilakukan untuk membangkitkan file proyek berekstensi *.qap yang berisi seluruh pengaturan halaman web termasuk data URL dan direktori keluaran.

e. Proses Pembangkitan Kode ASP

Proses ini dilakukan untuk membangkitkan file-file ASP beserta file-file pendukungnya (*Java Script*, *CSS*).



Gambar 3.4 DFD Level 2 Proses 1.1

Pada gambar di atas digambarkan DFD level 2 yang merupakan penjelasan dari proses 1.1 (Proses Pemetaan Isi dan Struktur Basis Data), yang terbagi menjadi dua sub proses:

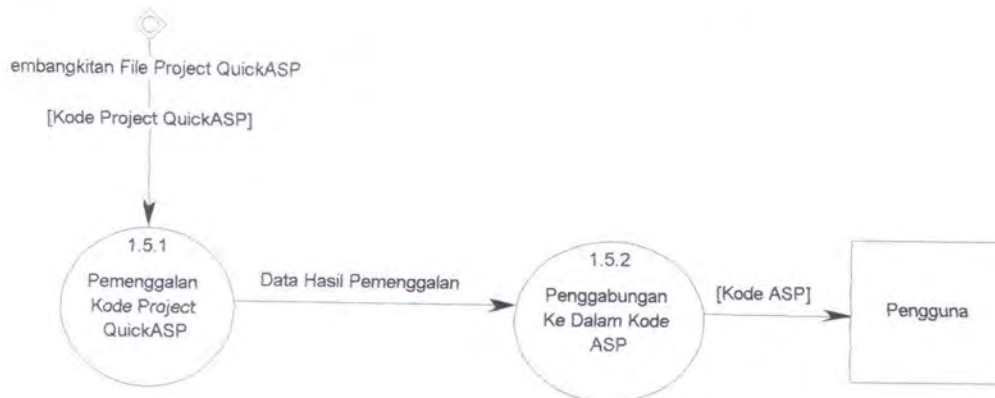
a. Proses Verifikasi Basis Data

Proses ini dilakukan untuk memeriksa apakah info basis data yang dimasukkan pengguna valid dan apakah basis data sudah ada di direktori *server*.

b. Proses Akses Basis Data

Jika status basis data valid, maka proses selanjutnya adalah mengakses basis data dan mendapatkan informasi mengenai isi basis data serta metadata basis data yang meliputi nama tabel, nama *field*, tipe data, panjang maksimum karakter, *primary key*, *foreign key*, *nullable field*, serta *identity field*.

Selanjutnya pada gambar di bawah terlihat gambar DFD level 2 yang merupakan penjelasan dari proses 1.5 (Pembangkitan Kode ASP).



Gambar 3.5 DFD Level 2 Proses 1.5

Proses pembangkitan kode ASP terbagi menjadi dua sub proses yaitu:

- Proses Pemenggalan Kode Proyek QuickASP

Setiap kali pengguna selesai melakukan pengaturan halaman web yang hendak dibangkitkan, aplikasi melakukan pembangkitan file proyek yang berekstensi *.qap untuk menyimpan hasil pengaturan tersebut. Selanjutnya file tersebut dibaca dan dilakukan pemenggalan (*parse*) isi file.

- Proses Penggabungan ke Dalam Kode ASP

Hasil pemenggalan isi file proyek tersebut selanjutnya dimasukkan ke dalam struktur data array guna mempermudah pada saat penggabungan dengan kode program ASP yang hendak dibangkitkan

3.3 Perancangan Tampilan Halaman Web

Perancangan tampilan halaman web yang dihasilkan ini diperlukan guna merancang form-form yang diperlukan untuk pembuatan perangkat lunak.

3.3.1 Halaman-Halaman Hasil Pembangkitan

Beberapa halaman standar yang dihasilkan oleh aplikasi QuickASP adalah:

- Halaman Daftar (*List Page*)

Merupakan halaman yang menampilkan barisan *record* pada suatu tabel dengan disertai fasilitas navigasi *record* yang dinamis. Biasanya tidak semua *field* ditampilkan pada halaman ini. Proses penghapusan *record* juga dapat dilakukan pada halaman ini.

- Halaman Penambahan Data (*Add Page*)

Yaitu halaman yang diperuntukkan bagi pengunjung yang ingin menambah *record* baru ke dalam suatu tabel basis data.

- Halaman Pengubahan Data (*Edit Page*)

Pada halaman ini pengunjung dapat mengubah *record* yang dikehendaki.

- Halaman Detail (*Detail Page*)

Berbeda dengan halaman daftar, tiap *record* ditampilkan secara vertikal dalam satu halaman dan biasanya lebih detail daripada halaman daftar.

Selain beberapa halaman standar di atas, aplikasi QuickASP juga menghasilkan halaman-halaman tambahan yang bersifat opsional yaitu:

- Halaman Login (*Login Page*)

Dengan adanya halaman ini, tingkat keamanan data dapat lebih terjaga. Setiap pengunjung memiliki hak akses yang berbeda sesuai dengan pengaturan halaman yang dilakukan oleh pengguna aplikasi QuickASP.

- Halaman Pencarian Data (*Search Page*)

Terdiri atas dua macam metode pencarian yaitu pencarian cepat (*Quick Search*) dan pencarian tingkat lanjut (*Advance Search*).

3.3.2 Tampilan Tipe Masukan dan Validasi Data

Untuk setiap tipe data yang berbeda, aplikasi QuickASP memberikan alternatif pilihan tipe masukan yang berbeda pula.

Berikut daftar tipe data beserta alternatif pilihan tipe masukan yang disediakan:

Tabel 3.1 Tipe Masukan Berdasarkan Tipe Data

MS Acces	Tipe Data		Pilihan Tipe Masukan
	MS SQL Server	Oracle	
<i>text, memo</i>	<i>char, nchar, ntext, nvarchar, text, varchar</i>	<i>char, nchar, nvarchar2, varchar2</i>	<i>textbox, combobox, listbox, radio button, checkbox</i>
<i>number, currency, autonumber</i>	<i>int, bigint, smallint, tinyint, binary, varbinary, decimal, float, money, smallmoney, numeric, real,</i>	<i>number, float</i>	<i>textbox, combobox, listbox, radio button</i>
<i>yes/no</i>	<i>bit</i>		<i>radio button, combobox, listbox</i>
<i>OLE object</i>	<i>image</i>	<i>long raw</i>	<i>input file</i>
<i>date/time</i>	<i>datetime, smalldatetime</i>	<i>date</i>	<i>datetimepicker, datetimeselector, textbox</i>

Khusus untuk tipe masukan *textbox*, aplikasi QuickASP juga memberikan pilihan validasi data bagi pengguna.

Jenis validasi data yang disediakan aplikasi dapat dilihat pada tabel di bawah ini :

Tabel 3.2 Jenis Validasi Data

Jenis Validasi	Format Validasi
Nomor telepon	Hanya boleh mengandung angka dan tanda penghubung “-“. Contoh: 021-345678
E-mail	Merupakan alamat e-mail yang valid. Contoh: xxx@xxx.xxx
Numerik	Hanya boleh mengandung angka
Desimal	Merupakan angka desimal yang valid (tanda pemisah desimal menggunakan tanda titik). Contoh: 56.78
Password	Tidak boleh kosong (harus diisi) dan minimal terdiri dari 4 karakter

Selain itu bagi *field* yang berjenis *not nullable field* (*field* yang harus diisi), aplikasi secara *default* memberikan validasi yang bersesuaian.

3.3.3 Sekuritas Pengunjung Situs Web

Selain tampilan yang menarik, sebuah situs web yang dinamis juga perlu didukung oleh tingkat keamanan data yang cukup tinggi. Salah satunya dengan memberikan batasan hak akses yang berbeda pada tiap pengunjung.

Alternatif pilihan yang diberikan kepada pengguna aplikasi QuickASP untuk menjaga keamanan data adalah:

- i. Tiap pengunjung hanya diperbolehkan melihat dan mengubah data-data miliknya
- ii. Tiap pengunjung diperbolehkan melihat dan mengubah data-data pengunjung lain
- iii. Tiap pengunjung diperbolehkan melihat data-data pengunjung lain namun hanya diperbolehkan mengubah data miliknya
- iv. Adanya fitur “*Login as Guest*” dimana jika pengunjung menggunakan login tersebut maka ia hanya bisa masuk ke halaman daftar (*List Page*)

- v. Adanya fitur "*Login as Admin*" dimana jika pengunjung memasukkan login dengan nama pengguna dan sandi tertentu yang diperlakukan sebagai admin dan telah tersimpan dalam basis data maka pengunjung dapat melakukan seluruh hal termasuk melihat, mengubah, dan menghapus data pada basis data.

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Tahap implementasi perangkat lunak dilakukan setelah tahap desain atau perancangan perangkat lunak selesai. Meskipun pada pelaksanaannya, akan ada proses dimana desain yang telah dibuat perlu diubah kembali untuk menyesuaikan dengan proses penulisan program. Pada bab ini akan dijelaskan komponen-komponen sistem yang membangun sistem QuickASP.

4.1 Lingkungan Implementasi

QuickASP ini dirancang dan dibuat dengan menggunakan perangkat keras dan perangkat lunak sebagai berikut :

1. Perangkat Keras
 - Komputer dengan Processor Intel Pentium III 598 MHz
 - Memori 512 Mb
 - Monitor Super VGA dengan resolusi 1024 x 768, warna 32 Bit
2. Perangkat Lunak
 - Sistem Operasi Windows XP Professional Version 2002
 - Internet Information System (IIS) sebagai *web server*
 - Microsoft Visual Basic 6.0 sebagai kakas untuk membuat perangkat lunak QuickASP
 - Macromedia DreamWeaver MX sebagai kakas untuk mendesain halaman web
 - Microsoft Access, Microsoft SQL Server, dan Oracle 9i sebagai basis data

4.2 Implementasi Antarmuka Pengguna QuickASP

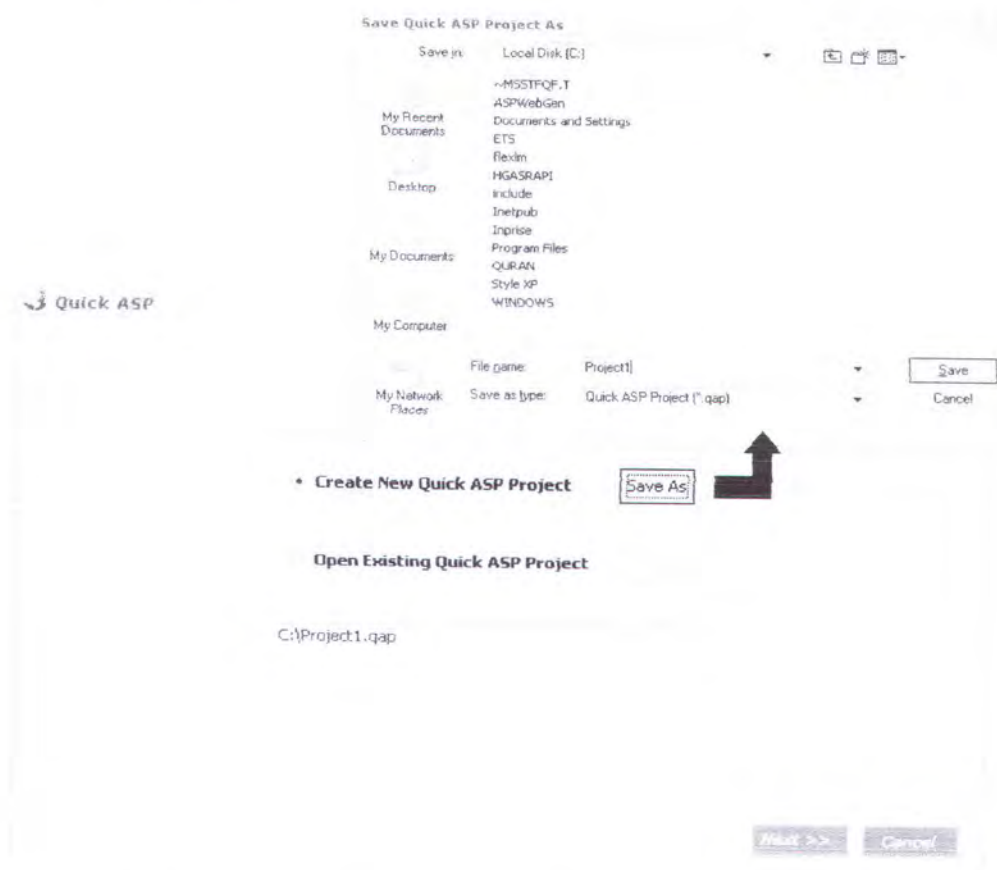
Aplikasi QuickASP merupakan pembangkit kode ASP yang menghasilkan sekumpulan halaman ASP dinamis yang mampu mengakses serta memodifikasi data pada basis data Microsoft Access, Microsoft SQL Server, dan Oracle 9i.

Aplikasi ini terdiri dari tiga bagian utama yaitu:

4.2.1 Bagian Form

Berikut ini form-form yang dijumpai oleh pengguna ketika menjalankan aplikasi QuickASP:

4.2.1.1 Form Pemilihan Proyek



Gambar 4.1 Form Pemilihan Proyek (pilihan pertama)

4.2.1.2 Form Pemilihan dan Pemasukan Info Basis Data

Selanjutnya pengguna harus memilih tipe basis data di antara ketiga pilihan yang disediakan aplikasi serta memasukkan nama basis data beserta properti lain yang diperlukan.

Ketiga pilihan tipe basis data tersebut adalah:

a. Microsoft Access

Gambar 4.3 Form Pemilihan dan Pemasukan Info Basis Data MS Access

Untuk mendapatkan file basis data Microsoft Access (*.mdb) yang hendak dibangkitkan, pengguna harus menekan tombol "Browse". Apabila diperlukan nama pengguna dan *password* untuk membuka basis data tersebut, maka pengguna juga harus mengisi kedua properti pada form yang disediakan.



b. *Microsoft SQL Server*

Quick ASP

- Microsoft SQL Server
- Microsoft Access
- Oracle

1. Select or enter a server name
ZAHRA
2. Enter information to log on to the server
 - ☐ Use Windows NT (Integrated Security)
 - ☐ Use a specific username and password
 - Username : sa
 - Password : xxxxxxxx
3. Select a database on the server
pubs

<< Back Next >> Cancel

Gambar 4.4 Form Pemilihan dan Pemasukan Info Basis Data SQL Server

Pada basis data MS SQL Server, selain mengisi nama server, pengguna juga harus memasukkan nama pengguna dan *password* jika diperlukan. Setelah itu pengguna harus memilih satu nama basis data yang hendak dibangkitkan

c. *Oracle*

Sama dengan pada basis data MS SQL Server, pada basis data Oracle selain mengisi nama server, pengguna juga harus memasukkan nama pengguna dan *password*.

The screenshot shows a web browser window titled "Quick ASP". On the left side, there is a list of database types: "Microsoft SQL Server", "Microsoft Access", and "Oracle". The "Oracle" option is selected, indicated by a bullet point. On the right side, there are two numbered steps for configuration. Step 1, "1. Enter a server name :", has an empty text input field below it. Step 2, "2. Enter information to log on to the database :", includes labels for "Username" and "Password", each followed by an empty text input field. At the bottom right of the form, there are three buttons: "<< Back", "Next >>", and "Cancel".

Gambar 4.5 Form Pemilihan dan Pemasukan Info Basis Data Oracle

Jika basis data tidak dapat dibuka maka aplikasi akan mengeluarkan pesan kesalahan.

4.2.1.3 Form Pemilihan Tabel dan Metode Pembangkitan

The screenshot shows a web application window titled "ASP Express". It contains two main sections:

- Choose table to generate ASP:** A list of tables with checkboxes:
 - ☒ authors
 - ☒ discounts
 - ☐ employee
 - ☐ jobs
 - ☐ pub_info
 - ☐ publishers
 - ☐ roysched
 - ☐ sales
 - ☐ stores
 - ☐ titleauthor
 - ☐ titles
 Below the list are two buttons: "Select All" and "Deselect All". At the bottom left of this section is a "Show Views" link.
- Select the way you build the site:** Two radio button options:
 - Automatically:** Automatically generates site without manually formatting.
 - Manually:** Generates site for each table step by step.

At the bottom right of the form are three buttons: "<< Back", "Next >>", and "Cancel".

Gambar 4.6 Form Pemilihan Tabel dan Metode Pembangkitan

Form ini terbagi menjadi dua bagian utama yaitu:

a. Bagian Pemilihan Tabel

Pada lajur kiri form ini, secara otomatis aplikasi akan menampilkan daftar tabel-tabel yang ada pada basis data yang telah dibuka. Selain daftar tabel, aplikasi juga dapat menampilkan daftar *view* pada basis data yang sama. Caranya pengguna harus menyalakan *checkbox* "Show views". Setelah tabel-tabel dan atau *view* yang hendak dibangkitkan dipilih, pengguna dapat bergeser pada bagian ke dua dari form ini yaitu bagian pemilihan metode pembangkitan.

b. Bagian Pemilihan Metode Pembangkitan

Ada dua macam metode pembangkitan yang ditawarkan oleh aplikasi QuickASP ini. Yang pertama adalah metode pembangkitan otomatis, sedangkan metode yang ke dua adalah metode pembangkitan manual.

Yang dimaksud dengan metode pembangkitan otomatis adalah metode pembangkitan kode ASP dimana pengguna tidak perlu melakukan pengaturan tampilan halaman web. Pengguna cukup memilih tabel-tabel yang hendak dibangkitkan, memilih *stylesheet* untuk tampilan halaman web secara keseluruhan, serta menentukan direktori keluaran. Tampilan halaman web yang dihasilkan diatur berdasarkan format pengaturan *default* untuk tiap *field*. Metode ini cocok bagi pengguna yang ingin mendapatkan halaman web ASP untuk beberapa tabel dalam waktu singkat.

Sedangkan pada metode pembangkitan yang ke dua, pengguna harus melakukan pengaturan format tampilan untuk tiap halaman web yang akan dihasilkan. Keunggulan dari metode ini adalah pengguna dapat mengganti format tampilan *default* yang telah disediakan aplikasi sesuai kehendaknya. Misalnya pada tipe masukan untuk tipe data tanggal (*datetime*). Secara *default*, aplikasi menggunakan tipe masukan *textbox* dengan fitur tambahan *datepicker* untuk mempermudah pengunjung web dalam mengisi tanggal. Sedangkan format tanggal secara *default* yang disediakan oleh QuickASP adalah format tanggal "dd/mmm/yyyy". Namun tipe masukan tersebut dapat diubah menjadi *date selector* yang berupa tiga buah *combobox*, masing-masing untuk tanggal, bulan, dan tahun. Pengguna juga dapat mengubah format tanggal menjadi "mmm/dd/yyyy". Namun dibandingkan dengan

metode pertama, metode ini membutuhkan waktu yang relatif lebih lama.

Apalagi jika pengguna hendak membangkitkan beberapa tabel sekaligus.

Selanjutnya pada setiap form pengaturan, nama tabel yang hendak dibangkitkan ditampilkan pada label bagian atas masing-masing form.

4.2.1.4 Form Pengaturan Tampilan Halaman Daftar (1)

Gambar 4.7 Form Pengaturan Tampilan Halaman Daftar (1)

Pengaturan tampilan halaman web yang pertama kali dilakukan oleh pengguna adalah pengaturan tampilan halaman daftar (*List Page*). Halaman daftar yang terbentuk nantinya berupa sebuah tabel yang menampilkan data beberapa atau seluruh *field*, fasilitas pencarian cepat (*Quick Search*), serta fasilitas navigasi *record*.

Form ini terdiri dari beberapa bagian:

- Pengatur urutan *field* pada halaman daftar

Untuk mengatur urutan *field*, sebelumnya pengguna harus memilih *field* apa saja yang hendak dicantumkan pada halaman daftar. Selanjutnya pengguna dapat mengatur urutannya dengan cara memilih *field* pada *listbox* sebelah kanan dan menggesernya ke atas atau ke bawah dengan menekan tombol anak panah atas atau bawah.

- Pengatur urutan data berdasarkan kategori tertentu baik terurut naik maupun terurut turun

Pengguna dapat menentukan data yang ditampilkan hendak diurutkan berdasarkan kategori *field* tertentu. Secara *default*, aplikasi mengurutkan data berdasarkan *field* yang pertama terurut naik. Untuk mengubah arah urutan datanya menjadi terurut turun, pengguna harus menyalakan *checkbox* “*Descending*”.

- Pengatur jumlah *record* tiap halaman (navigasi *record*)

Sebagaimana yang telah disebutkan di atas, salah satu fitur yang dibangkitkan oleh QuickASP pada halaman daftar adalah fitur navigasi *record*. Jumlah *record* yang dimasukkan oleh pengguna pada *textbox* “*Number of records on each page*” merupakan jumlah *record* pada saat pertama kali halaman daftar dibuka. Namun nantinya pada halaman daftar yang terbentuk, jumlah itu dapat berubah secara dinamis sesuai keinginan pengunjung web.

- Pengatur jumlah karakter maksimal yang ditampilkan pada tiap kolom

Untuk memperoleh tampilan halaman daftar yang lebih indah, pengguna dapat memanfaatkan fasilitas ini. Fasilitas ini berfungsi untuk membatasi jumlah karakter pada tiap kolom dari tabel yang ditampilkan pada halaman daftar hasil pembangkitan. Untuk melihat isi kolom tersebut secara keseluruhan, pengunjung dapat menekan *link "More"*.

4.2.1.5 Form Pengaturan Tampilan Halaman Daftar (2)

Pada form pengaturan tampilan halaman daftar yang ke dua ini, pengguna dapat melakukan pengaturan tampilan untuk masing-masing *field*. Form ini terdiri dari sebuah tabel dengan enam buah kolom di dalamnya.

Quick ASP

Shortcut

Project Option

DB Connection

User Guide

About Me

Customize your list page appearance, click the cell to edit it

Field	Label	List Page	Detail Page	View as	
discounttype	discounttype			Text	Wi
stor_id	stor_id	X		Text	Wi
lowqty	lowqty	X		Numeric	Wi
highqty	highqty	X		Numeric	Wi
discount	discount	X		Numeric	Wi

<< Back Next >> Cancel

Gambar 4.8 Form Pengaturan Tampilan Halaman Daftar(2)

Berikut penjelasan untuk masing-masing kolom:

a. *Field*

Yaitu kolom yang menampilkan seluruh nama *field* pada tabel yang hendak dibangkitkan

b. *Label*

Berfungsi untuk mengatur judul tiap kolom dari tabel (*head column*) pada halaman daftar yang dihasilkan. Aplikasi menggunakan nama *field* sebagai pengaturan *default*. Namun pengguna dapat mengubahnya dengan cara menekan sel yang hendak diubah.



Gambar 4.9 Pengaturan Kolom *Label* pada Form Pengaturan Halaman Daftar

c. *List Page*

Kolom ini menunjukkan apakah suatu *field* ditampilkan pada halaman daftar atau tidak. Pengisiannya tergantung dari pengisian pada form sebelumnya yaitu form pengaturan tampilan halaman daftar (1). Jika pada form sebelumnya suatu *field* dimasukkan ke dalam halaman daftar maka tanda yang tertera pada kolom ini adalah tanda “cek”. Jika sebaliknya maka tanda yang tertera adalah tanda “silang”. Untuk mengubah tandanya, pengguna harus kembali ke form sebelumnya.

d. *Detail Page*

Berbeda dengan kolom sebelumnya, kolom ini digunakan untuk mengatur apakah suatu *field* ditampilkan pada halaman detail atau tidak. Halaman detail adalah halaman yang menampilkan tiap *record* secara vertikal dalam satu halaman. Data-data pada halaman ini biasanya lebih detail daripada halaman daftar. Secara *default* aplikasi menampilkan seluruh *field* ke dalam halaman detail. Untuk mengubahnya pengguna tinggal menekan pada sel yang hendak diubah. Misal pengguna tidak ingin menampilkan *field* “stor_id” pada halaman detail maka pengguna harus mengubah tanda “cek” menjadi tanda “silang” pada baris *field* “stor_id” kolom “Detail Page”.



Gambar 4.10 Pengaturan Kolom *Detail Page* pada Form Pengaturan Halaman Daftar

e. *View As*

Kolom ini digunakan untuk melakukan pengaturan tampilan tiap *field* baik pada halaman daftar maupun pada halaman detail. Jenis tampilan yang disediakan tergantung pada tipe data serta jenisnya (*primary key* atau *foreign key*). Pengguna dapat mengubah jenis tampilan dengan menekan pada sel yang berkaitan sampai muncul *combobox* yang berisi pilihan jenis tampilan yang ditawarkan.

Field	View as
discounttype	Text
stor_id	Text
lowqty	Hyperlink
highqty	MailTo
discount	File-based Image

Tekan di sini

Gambar 4.11 Pengaturan Kolom *View As* pada Form Pengaturan Halaman Daftar

Berikut tipe data masing-masing basis data serta jenis tampilan yang ditawarkan:

Tabel 4.1 Jenis Tampilan pada Halaman Daftar dan Halaman Detail

Berdasarkan Tipe Data

MS Acces	Tipe Data		Pilihan Jenis Tampilan pada Halaman Daftar dan Halaman Detail
	MS SQL Server	Oracle	
<i>text, memo</i>	<i>char, nchar, ntext, nvarchar, text, varchar</i>	<i>char, nchar, nvarchar2, varchar2</i>	- Text - Hyperlink - MailTo - File-based Image
<i>number, autonumber</i>	<i>int, bigint, smallint, tinyint, binary, varbinary, decimal, float, numeric, real,</i>	<i>number, float</i>	- Numeric - Currency
<i>currency</i>	<i>money, smallmoney</i>		- Currency
<i>yes/no</i>	<i>bit</i>		- True/False
<i>OLE object</i>	<i>image</i>	<i>long raw</i>	- Database Image
<i>date/time</i>	<i>datetime, smalldatetime</i>	<i>date</i>	- ShortDate[Time] - LongDate[Time]

Penjelasan masing-masing jenis tampilan yang ditawarkan:

- a. *Text*

Data dengan format tampilan *text* akan ditampilkan apa adanya tanpa penambahan format apapun. Misal: string “saya” tetap ditampilkan menjadi “saya”.

b. *Hyperlink*

QuickASP akan mengubah data pada *field* tertentu menjadi sebuah *hyperlink*. Misal: “www.yahoo.com” ditampilkan menjadi “www.yahoo.com”

c. *MailTo*

Sama dengan *Hyperlink*, QuickASP akan mengubah data pada *field* tertentu menjadi sebuah *linking*. Misal: “quickasp@yahoo.com” ditampilkan menjadi “quickasp@yahoo.com”

d. *Numeric*

Seperti halnya format *Text*, data yang ditampilkan dengan format *Numeric* akan ditampilkan apa adanya. Misal: “23” tetap ditampilkan sebagai “23”

e. *Currency*

Data angka semisal 14000 akan ditampilkan sebagai “Rp 14.000,-” jika format mata uang yang dipilih adalah Indonesia. Sedangkan jika format mata uang yang dipilih adalah USA maka data tersebut akan ditampilkan sebagai “\$ 14,000”.

f. *True/False*

Data yang bertipe boolean akan ditampilkan dengan nilai yang terkandung di dalamnya, bukan semata-mata nilai “*True*” dan “*False*”. Misal pada *field* “Jenis_Kelamin” pengguna ingin menampilkannya dalam tulisan

“Pria” dan “Wanita”. Maka pengguna dapat melakukan pengaturan dengan mengisi nilai “Pria” pada *True Value* dan nilai “Wanita” pada *False Value*.

g. *Database Image*

Jika pengguna menyimpan gambar ke dalam basis data, maka QuickASP akan melakukan ekstraksi gambar tersebut dari basis data dan ditampilkan pada halaman daftar maupun halaman detail. Format gambar yang didukung adalah format JPEG, GIF, dan BMP.

h. *File-based Image*

Pilihan ini cocok jika pengguna ingin menampilkan gambar namun yang tersimpan dalam basis data adalah *file path* dari gambar tersebut.

i. *ShortDate[Time]*

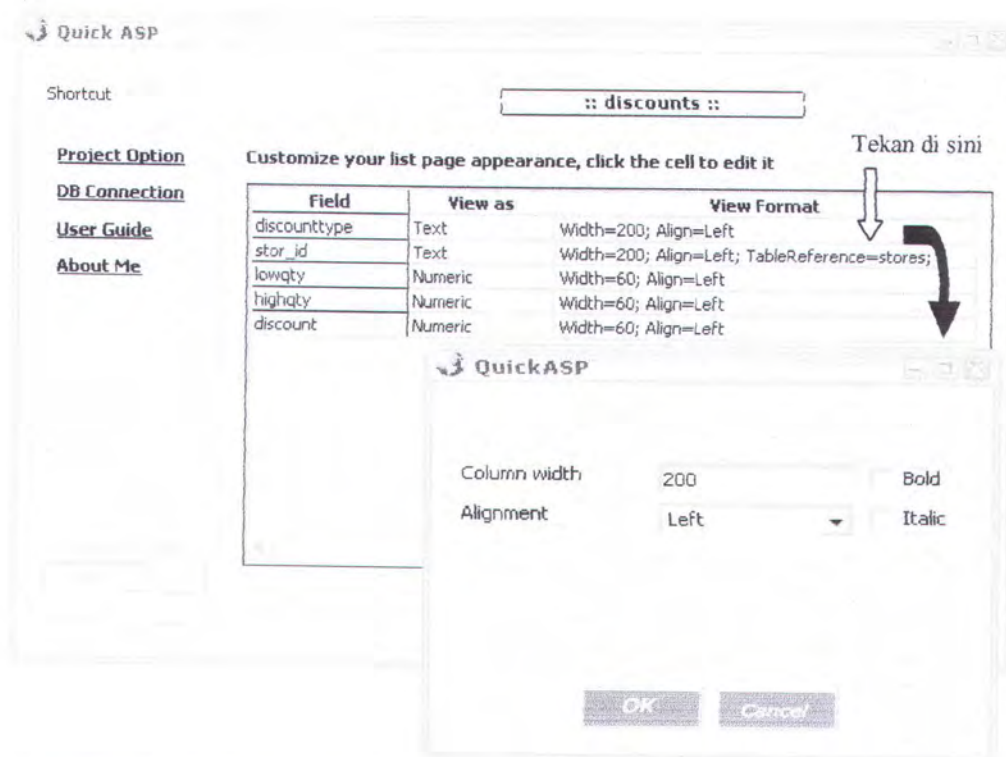
Menampilkan data dengan tipe tanggal (*datetime*) dengan format “dd/mm/yyyy” atau “mm/dd/yyyy”

j. *LongDate[Time]*

Menampilkan data dengan tipe tanggal (*datetime*) dengan format “dd/mmmm/yyyy” atau “mmmm/dd/yyyy”

f. *View Format*

Kolom ini berisi properti dari jenis masukan yang dipilih pada kolom “*View As*”. Pengguna dapat mengubah properti tersebut dengan cara menekan pada sel yang berkaitan sampai muncul form baru yang berisi pengaturan tampilan untuk tiap *field*.

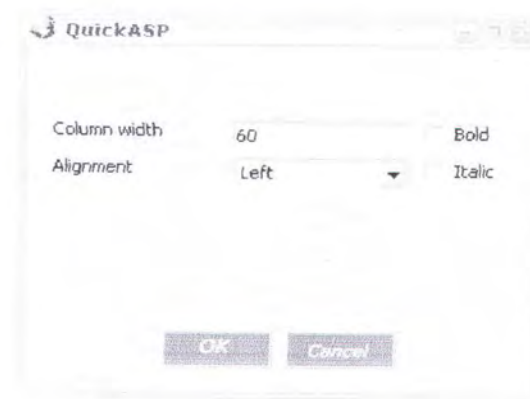


Gambar 4.12 Pengaturan Kolom *View Format* pada Form Pengaturan Halaman

Daftar

Tiap format tampilan yang dipilih pada kolom *View As* memiliki properti pengaturan yang berbeda.

- a. *Text, Hyperlink, MailTo, Numeric*



Gambar 4.13 Form Pengaturan Tampilan Tipe Data *Text*

Properti dari form ini:

☞ *Column width* : mengatur panjang kolom pada tabel halaman daftar

☞ *Alignment* : mengatur perataan kolom, terdiri dari 4 pilihan yaitu rata kiri, rata kanan, rata tengah, dan rata kiri kanan.

☞ *Bold* : mengatur tebal tidaknya teks pada kolom yang bersangkutan

☞ *Italic* : mengatur miring tidaknya teks pada kolom yang bersangkutan

b. Currency



Gambar 4.14 Form Pengaturan Tampilan Tipe Data *Currency*

☞ *Regional Setting* : mengatur format mata uang. Data angka semisal 14000 akan ditampilkan sebagai “Rp 14.000,-“ jika format mata uang yang

dipilih adalah Indonesia. Sedangkan jika format mata uang yang dipilih adalah USA maka data tersebut akan ditampilkan sebagai “\$ 14,000”.

Sedangkan properti *Column Width*, *Alignment*, *Bold*, dan *Italic* memiliki pengertian yang sama dengan properti pada format “*Text*”.

c. *True/False*

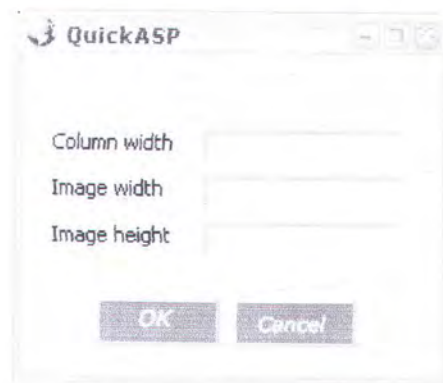
Gambar 4.15 Form Pengaturan Tampilan Tipe Data *True/False*

Properti dari form ini:

- ☞ *True Value (1)* : nilai yang ditampilkan jika *field* yang bersangkutan bernilai *True/Yes/1*
- ☞ *False Value (0)* : nilai yang ditampilkan jika *field* yang bersangkutan bernilai *False/No/0*

Sedangkan properti *Column Width*, *Alignment*, *Bold*, dan *Italic* memiliki pengertian yang sama dengan properti pada format “*Text*”.

- d. Database Image, File-based Image



Gambar 4.16 Form Pengaturan Tampilan Tipe Data *Image*

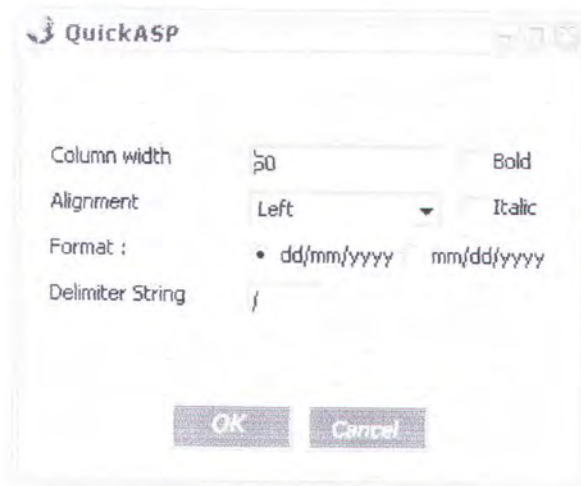
Properti dari form ini:

∞ *Image Width* : mengatur lebar gambar

∞ *Image Height* : mengatur tinggi gambar

Sedangkan properti *Column Width* memiliki pengertian yang sama dengan properti pada format "*Text*".

- e. ShortDate[Time], LongDate[Time]



Gambar 4.17 Form Pengaturan Tampilan Tipe Data *Date*

Properti dari form ini:

- ☞ *Format* : mengatur format tampilan tanggal
(dd/mm/yyyy atau mm/dd/yyyy)
- ☞ *Delimiter String* : menentukan tanda pemisah antara tanggal,
bulan, dan tahun. Nilai *default* adalah tanda
“/”.

Khusus bagi *field* yang berkedudukan sebagai *foreign key* maka aplikasi menambahkan fasilitas untuk melakukan pengaturan *field* apa pada tabel master yang hendak dijadikan tampilan (*display field*).

4.2.1.6 Form Pengaturan Tampilan Halaman Ubah (1)



Gambar 4.18 Form Pengaturan Tampilan Halaman Ubah (1)

Pada form ini pengguna dapat melakukan pengaturan urutan *field* yang hendak ditampilkan pada halaman ubah hasil pembangkitan. Untuk mengubah urutan, pengguna harus memilih salah satu *field* dan menggesernya ke atas atau ke bawah dengan cara menekan tombol panah atas atau panah bawah.

4.2.1.7 Form Pengaturan Tampilan Halaman Ubah (2)

Sebagaimana pada form pengaturan tampilan halaman daftar (2), pada form ini pengguna juga dapat melakukan pengaturan tampilan untuk masing-masing *field*. Form ini terdiri dari sebuah tabel dengan enam buah kolom di dalamnya.

Shortcut: **discounts**

Project Option
DB Connection
User Guide
About Me

Customize your detail and edit page appearance, click the cell to edit it

Field	Label	Edit	Add	Edit as	
discounttype	discounttype			Text Field	Size=60; MaxLength=40; Tipe=Tex
stor_id	stor_id			Combo Box	TableParse(Display,Value):stores(st
lowqty	lowqty			Text Field	Size=15; MaxLength=50; Tipe=Tex
highqty	highqty			Text Field	Size=15; MaxLength=50; Tipe=Tex
discount	discount			Text Field	Size=15; MaxLength=50; type=Te

<< Back Next >> Cancel

Gambar 4.19 Form Pengaturan Tampilan Halaman Ubah (2)

Berikut penjelasan untuk masing-masing kolom:

a. *Field*

Yaitu kolom yang menampilkan seluruh nama *field* pada tabel yang hendak dibangkitkan

b. *Label*

Berfungsi untuk mengatur judul tiap kolom dari tabel (*head column*) pada halaman ubah yang dihasilkan. Aplikasi menggunakan nama *field* sebagai pengaturan *default*. Namun pengguna dapat mengubahnya dengan cara menekan sel yang hendak diubah. Cara mengubahnya sama dengan petunjuk pada form pengaturan tampilan halaman daftar.

c. *Edit Page*

Kolom ini menunjukkan apakah suatu *field* ditampilkan pada halaman ubah (*Edit Page*) atau tidak. Tanda “cek” menunjukkan bahwa suatu *field* ditampilkan pada halaman ubah. Sebaliknya, tanda “silang” menunjukkan bahwa *field* tidak ditampilkan pada halaman ubah. Untuk mengubah tandanya, pengguna dapat menekan pada sel yang bersesuaian. Secara *default*, aplikasi menampilkan seluruh *field* pada halaman ubah kecuali bagi *field* yang berfungsi sebagai *Identity Field*. Hal ini disebabkan *field* ini memiliki sifat terisi secara otomatis (*autoinsert*). Untuk mengubahnya, pengguna tinggal menekan tanda pada sel yang bersangkutan.

d. *Add Page*

Kolom ini digunakan untuk mengatur apakah suatu *field* ditampilkan pada halaman tambah atau tidak. Sebagaimana pada kolom *Edit Page*, secara *default* aplikasi menampilkan seluruh *field* pada halaman tambah kecuali bagi

field yang berfungsi sebagai *Identity Field*. Untuk mengubahnya, pengguna tinggal menekan tanda pada sel yang bersangkutan.

e. *Edit As*

Kolom ini digunakan untuk melakukan pengaturan tampilan tiap *field* baik pada halaman ubah maupun pada halaman tambah. Jenis tampilan berupa kotak masukan yang tergantung pada tipe data serta jenisnya (*primary key* atau *foreign key*).

Berikut tipe data serta jenis tampilan yang ditawarkan:

Tabel 4.2 Jenis Masukan pada Halaman Ubah dan Halaman Tambah
Berdasarkan Tipe Data

MS Acces	Tipe Data		Pilihan Jenis Masukan pada Halaman Ubah dan Halaman Tambah
	MS SQL Server	Oracle	
<i>text, memo</i>	<i>char, nchar, ntext, nvarchar, text, varchar</i>	<i>char, nchar, nvarchar2, varchar2</i>	<ul style="list-style-type: none"> - <i>Text field</i> - <i>Text area</i> - <i>Radio button</i> - <i>Combobox</i> - <i>Listbox</i> - <i>Checkbox</i> - <i>File-based image</i>
<i>number, autonumber, currency</i>	<i>int, bigint, smallint, tinyint, binary, varbinary, decimal, float, numeric, real, money, smallmoney</i>	<i>number, float</i>	<ul style="list-style-type: none"> - <i>Text field</i> - <i>Radio button</i> - <i>Combobox</i> - <i>Listbox</i>
<i>yes/no</i>	<i>bit</i>	-	<ul style="list-style-type: none"> - <i>Radio button</i> - <i>Combobox</i> - <i>Listbox</i>
<i>OLE object</i>	<i>image</i>	<i>long raw</i>	- <i>Database Image</i>
<i>date/time</i>	<i>datetime, smalldatetime</i>	<i>date</i>	- <i>Date</i>

Khusus bagi *field* yang berkedudukan sebagai *foreign key* maka aplikasi hanya memberikan pilihan jenis masukan yaitu *combobox*, *listbox*, dan *radio button*. Hal ini dimaksudkan supaya data yang tercantum pada ketiga tipe masukan tersebut hanya data dari tabel master.

f. *Edit Format*

Kolom ini berisi properti dari jenis masukan yang dipilih pada kolom “*Edit As*”. Pengguna dapat mengubah properti tersebut dengan cara menekan pada sel yang berkaitan sampai muncul form baru yang berisi pengaturan tampilan untuk tiap *field*.

Berikut gambar form pengaturan beserta propertinya untuk tiap-tiap jenis masukan:

▪ *Text Field*

QuickASP

au_id

Size : (char) 60

Max Length : (char) 11

Password Field

Required Field

Validate as :

Default Value :

No Validation

No Validation

Phone Number(###-###)

E-mail(*@*.*)

Number(###)

Decimal(##.##)

Cancel

Gambar 4.20 Form Pengaturan Jenis Masukan *Text Field*

Properti dari form ini:

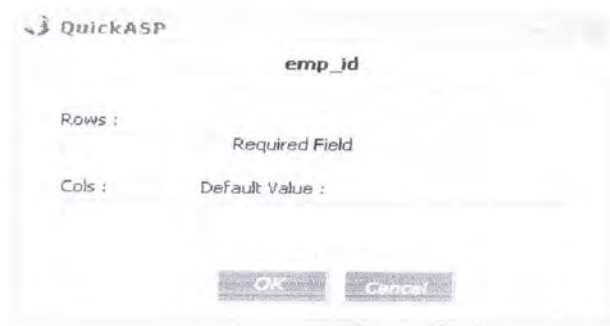
Size : ukuran lebar *textfield* (dalam karakter)

- ☞ *Max Length* : jumlah maksimal karakter yang dimasukkan
- ☞ *Validate as* : pilihan jenis validasi, dapat dilihat pada tabel 4.3
- ☞ *Default Value* : nilai *default* dari *textfield*
- ☞ *Password Field* : untuk mengatur apakah *field* yang bersangkutan termasuk jenis *password*.
Jika ya maka aplikasi akan membuat validasinya
- ☞ *Required Field* : untuk mengatur apakah *field* yang bersangkutan harus diisi. Jika ya maka aplikasi akan membuat validasinya

Tabel 4.3 Jenis Validasi Data pada Pengaturan Tampilan Halaman Ubah dan Halaman Tambah

Jenis Validasi	Format Validasi
<i>Phone Number</i>	Hanya boleh mengandung angka dan tanda penghubung “-”. Contoh: 021-345678
<i>E-mail</i>	Merupakan alamat e-mail yang valid. Contoh: xxx@xxx.xxx
<i>Numeric</i>	Hanya boleh mengandung angka
<i>Decimal</i>	Merupakan angka desimal yang valid (tanda pemisah desimal menggunakan tanda titik). Contoh: 56.78

▪ Text Area



Gambar 4.21 Form Pengaturan Jenis Masukan *Text Area*

Properti dari form ini:

- ☞ *Rows* : jumlah baris pada *text area*
- ☞ *Cols* : jumlah kolom pada *text area*
- ☞ *Required Field* : untuk mengatur apakah *field* yang bersangkutan harus diisi. Jika ya maka aplikasi akan membuatkan validasinya
- ☞ *Default Value* : nilai *default* dari *text area*

▪ *Radio Button, Listbox, Combobox, Checkbox*

Pada keempat jenis masukan ini, aplikasi menyediakan dua pilihan pengaturan, yaitu:

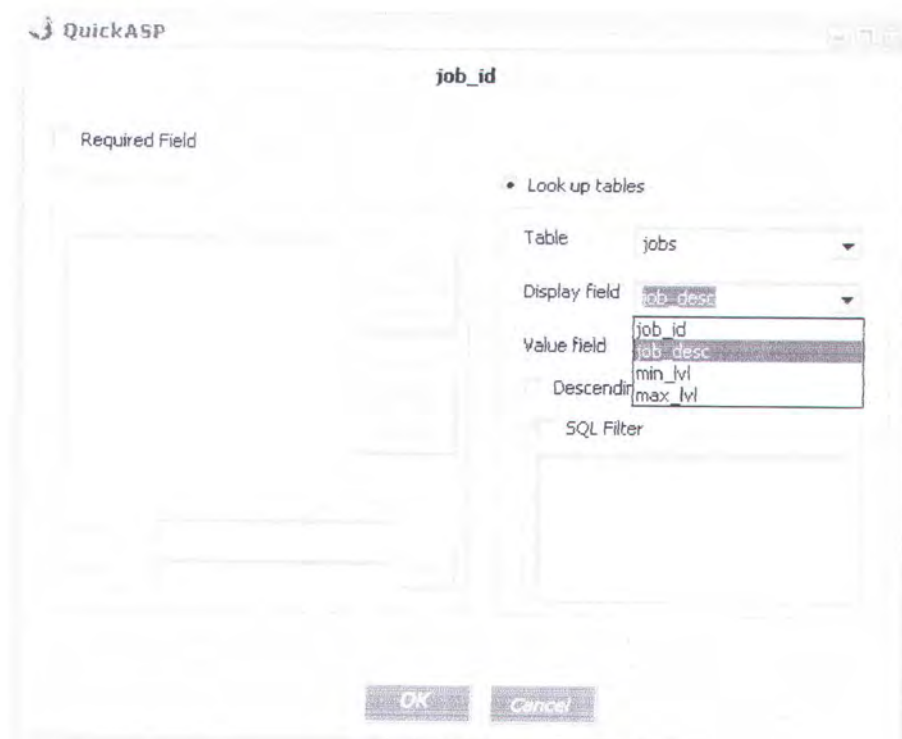
- Pemasukan data dari pengguna (*List of Values*)



Gambar 4.22 Form Pengaturan Jenis Masukan *Combobox* (Pilihan Pertama)

Pengguna dapat mengatur secara manual pilihan apa saja yang hendak dimasukkan ke dalam *radio button*, *listbox*, *combobox*, maupun *checkbox* hasil pembangkitan nantinya. Untuk menambahkan data baru, pertama kali pengguna harus menekan tombol “Add”. Kemudian mengisi nilai tampilan dan nilai sebenarnya dari pilihan-pilihan yang hendak dimasukkan pada *combobox* hasil pembangkitan. Pengguna dapat mengatur urutan pilihan tersebut dengan cara menekan tombol “Up” atau “Down”.

- Pemasukan data dari tabel (*Look Up Tables*)

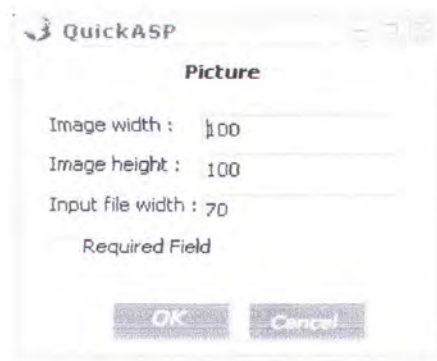


Gambar 4.23 Form Pengaturan Jenis Masukan *Combobox* (Pilihan Kedua)

Untuk memudahkan pengguna, pilihan-pilihan pada *combobox* hasil pembangkitan juga dapat diambil dari tabel tertentu. Pengguna cukup memilih tabel serta *field* yang hendak ditampilkan. Selain dengan memilih tabel, pengguna juga dapat mengisi sintaks SQL yang valid ke dalam *SQL Filter*.

- *File-based Image, Database Image*

Untuk memasukkan image maupun *path image* ke dalam basis data, digunakan jenis masukan *input file*.



Gambar 4.24 Form Pengaturan Jenis Masukan *File-based Image* dan *Database*

Image

Properti dari form ini:

- ∞ *Image Width* : lebar gambar
- ∞ *Image Height* : tinggi gambar
- ∞ *Input File Width* : lebar *input file*

▪ Date

Ada dua pilihan jenis masukan *Date*:

- i. *Textbox*
- ii. *Dropdown Combobox*

Pengguna juga dapat memilih format tanggal yang dikehendaki. Empat pilihan format tanggal yang disediakan aplikasi yaitu:

- i. dd/mm/yyyy
- ii. dd/mmm/yyyy
- iii. mm/dd/yyyy
- iv. mmm/dd/yyyy

Selain tanggal, pengguna juga bisa menambahkan format waktu. Dua pilihan format waktu yaitu:

- i. hh:mm:ss AM/PM
- ii. hh:mm:ss

Gambar 4.25 Form Pengaturan Jenis Masukan Date

Properti lain dari form ini adalah:

☞ *Datepicker*



Yaitu fitur untuk mengambil tanggal, bulan dan tahun dari sebuah fungsi kalender secara cepat



Gambar 4.26 Date Time Picker

∞ Year Range

Jika pengguna memilih jenis masukan *Dropdown Combobox* maka pengguna juga harus memasukkan rentang tahun yang akan dipilih.

∞ Default Value

Nilai *default* untuk jenis masukan *Date*. Nilai *default* tersebut harus berupa format tanggal yang valid atau berupa fungsi tanggal seperti *Now()*.

4.2.1.8 Form Pengaturan Tampilan Halaman Pencarian

Shortcut:

[Project Option](#)
[DB Connection](#)
[User Guide](#)
[About Me](#)

Customize your search page appearance, click the cell to edit it

Field	Search	Advanced Search	Search Format
discounttype	<input type="checkbox"/>	<input type="checkbox"/>	unspecified format
stor_id	<input type="checkbox"/>	<input type="checkbox"/>	Selection style=Combobox; Table=stores; Va
lowqty	<input type="checkbox"/>	<input type="checkbox"/>	unspecified format
highqty	<input type="checkbox"/>	<input type="checkbox"/>	unspecified format
discount	<input type="checkbox"/>	<input type="checkbox"/>	unspecified format

<< Back Next >> Cancel

Gambar 4.27 Form Pengaturan Tampilan Halaman Pencarian

Form ini terbagi menjadi empat kolom:

1. *Field*

Yaitu kolom yang menampilkan seluruh nama *field* pada tabel yang hendak dibangkitkan

2. *Quick Search*

Yaitu kolom untuk melakukan pengaturan apakah suatu *field* ditampilkan pada kategori pencarian cepat atau tidak. Disebut pencarian cepat karena pengguna cukup memilih kategori pencarian serta memasukkan kata kunci pencarian. Fasilitas ini terdapat pada bagian atas halaman daftar. Jika ingin memasukkan suatu *field* ke dalam kategori pencarian cepat, maka pengguna harus mengubah tanda “silang” menjadi tanda “cek”. Begitu pula sebaliknya.

3. Advance Search

Kolom ini digunakan untuk mengatur apakah suatu *field* dimasukkan ke dalam kategori pencarian tingkat lanjut atau tidak.

4. Search Format

Bagi *field* yang termasuk dalam kategori *foreign key*, maka aplikasi secara otomatis mengisi kolom ini dengan nama tabel induk serta *field* yang berelasi pada tabel induk. Sedangkan bagi *field* lainnya, aplikasi tidak melakukan pengaturan khusus.

4.2.1.9 Form Pengaturan Tampilan Halaman Login

Gambar 4.28 Form Pengaturan Tampilan Halaman Login

Halaman login bersifat opsional. Jika pengguna ingin agar aplikasi juga membangkitkan halaman login untuk tiap tabel yang hendak dibangkitkan maka pengguna dapat menyalakan *checkbox* “Create Login Page”.

Ada dua macam halaman login yang dapat dipilih oleh pengguna berdasarkan jenis nama pengguna (*username*) dan sandi (*password*) yang dimasukkan:

- a. Halaman login yang menggunakan nama pengguna dan sandi yang sama untuk seluruh pengunjung web.
- b. Halaman login yang mengambil nama pengguna dan sandi dari tabel suatu basis data. Untuk itu pengguna harus memilih nama tabel tempat menyimpan nama pengguna dan sandinya serta memilih *field* yang berisi data nama pengguna dan *field* yang berisi data sandi. Selain itu terdapat fasilitas tambahan yaitu pengaturan sekuritas tingkat lanjut sebagaimana yang terlihat pada gambar 4.17

Quick ASP

:: discounts ::

Customize users privileges

Users can see and edit other users data

- Users can see and edit their own data only

Users can see other users data; can edit their own data only

Users Table: OwnerID Field
emp_id

Main Table: OwnerID Field
stor_id

✓ Add admin user that can see and edit all users data
Admin UserID PSA89086M - Pedro

✓ Add "Login as Guest" with readonly access

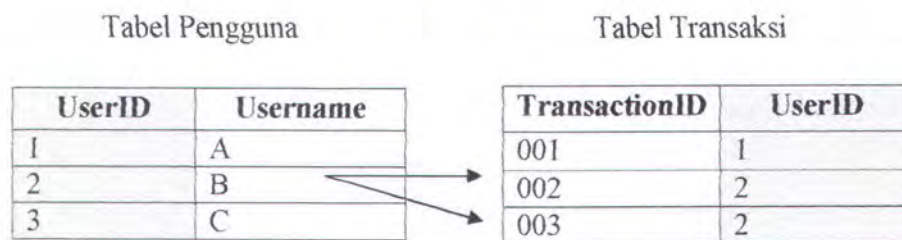
OK Reset

Gambar 4.29 Form Pengaturan Tampilan Halaman Login (Tingkat Lanjut)

Ada tiga macam hak akses yang dapat dipilih pengguna:

1. Setiap pengunjung yang berhasil login dapat melihat, mengubah, dan menghapus data pengunjung lain
2. Setiap pengunjung yang berhasil login hanya dapat melihat, mengubah, dan menghapus datanya sendiri.
3. Setiap pengunjung yang berhasil login dapat melihat data pengunjung lain namun hanya dapat mengubah dan menghapus datanya sendiri.

Untuk pilihan hak akses ke-2 dan ke-3 pengunjung harus memilih salah satu *field* pada tabel yang berisi nama pengguna dan sandi dan salah satu *field* pada tabel yang sedang dibangkitkan. Kedua *field* yang dipilih tersebut memiliki hubungan relasi yang dapat digambarkan sebagai berikut:



Dengan memilih hak akses sekuriti ke-2 dan ke-3, Username B hanya memiliki hak akses untuk TransactionID 002 dan 003.

Selain ketiga pilihan hak akses tersebut, aplikasi QuickASP juga menyediakan fasilitas login untuk Admin yang dapat melihat, mengubah, serta menghapus data semua pengunjung. Juga terdapat fasilitas "*Login as Guest*" dimana pengunjung yang masuk dengan login ini hanya bisa melihat data namun tidak dapat menambah, mengubah, dan menghapus data. Kedua fasilitas ini hanya bisa diaktifkan jika pengguna memilih sekuritas hak akses ke-2 atau ke-3.

4.2.1.10 Form Pemilihan *Stylesheet* dan Direktori Keluaran

The screenshot shows a web browser window titled 'QuickASP'. At the top, there is a navigation bar with 'Shortcut' and ':: authors ::'. Below this, there are links for 'Project Option', 'DB Connection', 'User Guide', and 'About Me'. The main content area is divided into two sections: 'Web style' and 'Output directory'. In the 'Web style' section, there is a dropdown menu for 'Stylesheet' with 'Nature.css' selected, and a 'Stylesheet preview' area below it. In the 'Output directory' section, there is a text input field for 'Browse for output directory' with the value 'C:\inetpub\wwwroot\pr1\' and a 'Browse' button next to it. Below this is a 'Web Site URL' input field with the value 'http://localhost/'. At the bottom of the form, there are three buttons: '<< Back', 'Next >>', and 'Cancel'. A 'GO!' button is also visible in the bottom left corner of the form area.

Gambar 4.30 Form Pemilihan *Stylesheet* dan Direktori Keluaran

Form ini terdiri dari dua bagian utama yaitu bagian pemilihan *stylesheet* dan bagian pemilihan direktori keluaran.

a. Bagian Pemilihan *Stylesheet*

Stylesheet merupakan file CSS yang secara umum digunakan untuk mengatur tampilan *homepage* ketika dilihat dari *browser*. Pada form ini pengguna dapat memilih *stylesheet* yang disediakan oleh aplikasi yaitu:

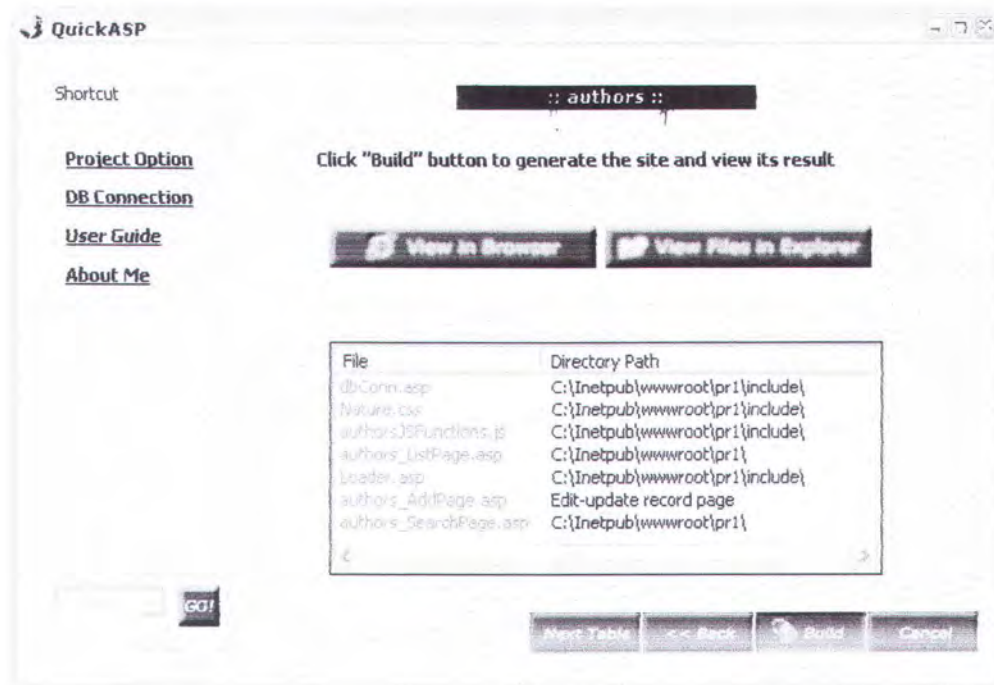
- i. Ocean.css
- ii. Nature.css
- iii. Elegant.css
- iv. Sunkist.css
- v. Rose.css

Tampilan dari tiap-tiap *stylesheet* dapat dilihat pada *Stylesheet Preview*.

b. Bagian Pemilihan Direktori Keluaran

Pengguna harus menentukan direktori keluaran untuk menyimpan hasil pembangkitan berupa file ASP beserta file-file pendukungnya. Selain itu pengguna juga dapat mengisi URL dari web yang terbentuk sebagai alamat awal ketika pengguna ingin melihat hasil pembangkitan pada *browser*.

4.2.1.11 Form Pembangkitan Kode Program ASP



Gambar 4.31 Form Pembangkitan Kode ASP

Setelah seluruh pengaturan selesai dilakukan, pengguna dapat menekan tombol "Build" yang akan membangkitkan file-file ASP beserta file-file pendukung lainnya. Namun jika ternyata masih ada tabel yang sudah dipilih oleh pengguna pada form pemilihan tabel dan belum dilakukan pengaturan tampilannya, maka aplikasi akan menanyakan terlebih dahulu kepada pengguna

apakah hendak melanjutkan proses pembangkitan atau melanjutkan ke pengaturan tampilan untuk tabel berikutnya. Jika pengguna ingin melanjutkan ke pengaturan untuk tabel yang lain maka pengguna harus menekan tombol “*Next Table*” dan secara otomatis aplikasi kembali pada form pemilihan tabel. Pada form ini terdapat *listbox* yang menampilkan halaman-halaman yang dihasilkan oleh QuickASP.

Sesudah aplikasi QuickASP berhasil membangkitkan file ASP dan file-file pendukung lainnya ke dalam direktori keluaran yang telah dipilih pengguna, pengguna diberi dua macam pilihan untuk melihat hasil pembangkitan tersebut.

Pertama, pengguna dapat melihat hasil pembangkitan secara langsung pada browser (*Internet Explorer*). Alamat yang dituju pertama kali adalah halaman web yang dianggap sebagai halaman *default*.

Kedua, pengguna juga dapat melihat file-file hasil pembangkitan lewat *Windows Explorer*.

File-file yang dihasilkan oleh QuickASP yaitu:

- a. default.asp

Merupakan halaman yang dituju pengunjung pertama kali

- b. <NAMA_TABEL>_ListPage.asp

Berisi daftar *record* dari suatu tabel yang ditampilkan secara horisontal, fasilitas pencarian cepat, serta navigasi *record*

- c. <NAMA_TABEL>_EditPage.asp

Terdiri dari tiga halaman utama yaitu halaman penambahan record baru, halaman pengubahan record, serta halaman detail.

d. <NAMA_TABEL>_SearchPage.asp

Berisi fasilitas pencarian data tingkat lanjut.

e. <NAMA_TABEL>_Login.asp

Merupakan halaman login bagi tiap pengunjung web

f. include/<NAMA_TABEL>_JSFunctions.js

Berisi fungsi-fungsi *Java Script* yang mendukung halaman-halaman ASP

g. include/Imager.asp

Fungsi untuk menampilkan gambar-gambar dari basis data

h. include/Loader.asp

Fungsi untuk memasukkan gambar ke database

i. include/Datetimepicker.js

Fungsi *Java Script* untuk membuat *datepicker*

j. include/Datetimedropdown.asp

Fungsi ASP untuk validasi *date-dropdown*

k. include/Datetimedropdown.js

Fungsi *Java Script* untuk validasi *date-dropdown*

l. include/LoginValidation.js

Fungsi *Java Script* untuk validasi halaman login

m. include/dbConn.asp

Berisi koneksi string basis data

n. include/xxx.css

Stylesheet untuk halaman web yang terbentuk

4.2.2 Bagian Tombol



Gambar 4.32 Macam-macam Tombol pada QuickASP

Selain terdiri dari berbagai form utama, aplikasi QuickASP juga memiliki beberapa tombol yang terletak pada bagian bawah form utama.

Masing-masing tombol tersebut adalah:

- Tombol "*Back*"

Tombol ini berfungsi jika pengguna ingin kembali ke form sebelumnya.

- Tombol "*Next*"

Tombol ini berfungsi jika pengguna ingin melanjutkan pengaturan ke form berikutnya.

- Tombol "*Cancel*"

Tombol ini berfungsi jika pengguna ingin mengakhiri proses pembangkitan. Namun sebelum pengguna benar-benar keluar dari aplikasi, terlebih dulu aplikasi melakukan konfirmasi apakah benar pengguna ingin mengakhiri proses pembangkitan.

- Tombol "*Build*"

Tombol ini hanya terdapat pada dua form yaitu pada form pemilihan proyek ketika pengguna memilih untuk membuka proyek yang sudah ada serta pada form pembangkitan kode program. Dengan menekan tombol "*Build*" ini, aplikasi akan membangkitkan file proyek QuickASP serta file-file ASP dan file-file pendukungnya pada direktori yang telah dipilih pengguna.

- Tombol "*Modify*"

Tombol ini hanya terdapat pada form pemilihan proyek ketika pengguna memilih untuk membuka proyek yang sudah ada. Dengan menekan tombol ini maka pengguna dapat melihat tahap demi tahap pengaturan tampilan halaman yang akan dibangkitkan berdasarkan proyek QuickASP yang sedang dibuka.

- Tombol "*Next Table*"

Tombol ini hanya terdapat pada form pembangkitan kode program ASP dan hanya terlihat ketika pengguna memilih menggunakan metode pembangkitan manual untuk lebih dari satu tabel. Setelah menekan tombol ini maka aplikasi akan membawa pengguna menuju form pemilihan tabel untuk memilih tabel mana yang hendak dibangkitkan berikutnya.

4.2.3 Bagian Shortcut

Untuk efisiensi dan kemudahan dalam penggunaan aplikasi ini, aplikasi QuickASP juga dilengkapi dengan beberapa *shortcut* yang terdapat pada semua form kecuali form pemilihan proyek, form pemilihan dan pemasukan info basis data, serta form pemilihan tabel dan metode pembangkitan.

Shortcut yang disediakan yaitu:

- *Shortcut "Project Option"*

Membawa pengguna kepada form pemilihan proyek.



Gambar 4.33 *Shortcut* pada QuickASP

- *Shortcut “DB Connection”*

Membawa pengguna kepada form pemilihan dan pemasukan info basis data.

- *Shortcut “User Guide”*

Membantu pengguna dalam penggunaan aplikasi QuickASP.

- *Shortcut “About Me”*

Informasi umum mengenai aplikasi QuickASP.

- *Shortcut “Previous Table”*

Shortcut ini memberikan kemudahan bagi pengguna untuk kembali pada pengaturan tabel-tabel sebelumnya. Caranya dengan memilih tabel yang hendak dituju pada *combobox* yang tersedia lalu menekan tombol “Go”.

- *Shortcut “Start New Project”*

Shortcut ini hanya terdapat pada form pembangkitan kode ASP. Digunakan ketika pengguna ingin kembali pada form awal dan memulai proyek baru.

BAB V

UJI COBA DAN EVALUASI

Sistem yang telah diimplementasikan akan diuji coba pada tahap ini untuk mengetahui apakah proses-proses yang telah dirancang dapat berjalan dengan baik.

Uji coba tersebut dilakukan untuk dengan tujuan:

1. Mengetahui apakah QuickASP dapat membangkitkan file-file ASP beserta file-file pendukungnya
2. Apakah halaman web hasil pembangkitan QuickASP dapat melakukan proses menambah, menghapus, mengubah data serta proses menu navigasi.

5.1 Lingkungan Uji Coba

Lingkungan uji coba adalah komputer-komputer tempat dimana uji coba sistem dan aplikasi dilakukan. Tiap-tiap komputer mempunyai spesifikasi khusus agar uji coba dilakukan dengan lancar. Dalam uji coba ini dibedakan menjadi dua bagian yaitu *server* dan *client*.

5.1.1 Server

Server yang digunakan untuk uji coba mempunyai spesifikasi perangkat keras dan perangkat lunak sebagai berikut :

♦ Server Lokal

- Prosessor Pentium 3 598 MHz
- RAM 512 Mb

- Hard disk 20 Gb
- LAN Card
- Operating system Windows XP Profesional 2002
- Internet Information Services (IIS) sebagai *server*
- Internet Explorer 5.5 keatas

♦ Server FTP

- Prosessor Pentium 4 2,50 GHz
- RAM 112 Mb
- Hard disk 10 Gb
- LAN Card
- Operating system Windows XP Profesional 2002
- Internet Information Services (IIS) sebagai *web server*
- Internet Explorer 5.5 keatas

5.1.2 Client

Pada uji coba ASPWebGen, *client* yang digunakan mempunyai spesifikasi perangkat keras dan perangkat lunak sebagai berikut :

- Prosessor Pentium 3 598 MHz
- RAM 512 Mb
- Hard disk 40 Gb
- LAN Card
- Operating system Windows XP Profesional 2002
- Internet Information Services (IIS) sebagai *web server*

- ASPWebGen
- Internet Explorer 5.5 keatas

5.2 Skenario Uji Coba

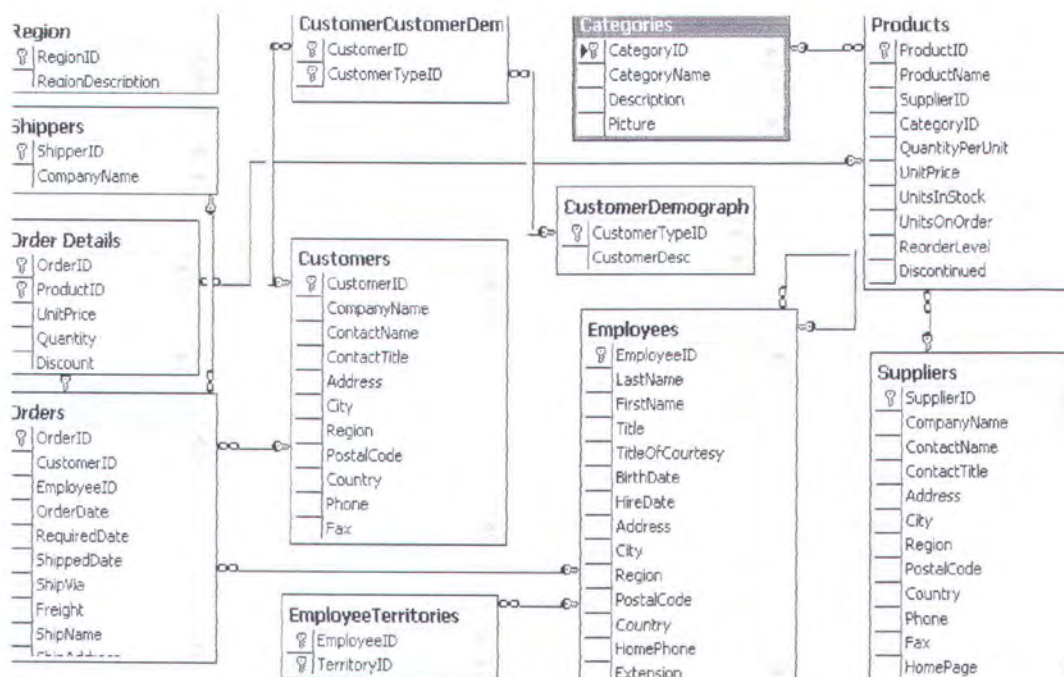
Berikut ini dilakukan beberapa uji coba dengan masukan basis data dan file CSS yang berbeda dengan tujuan untuk mengetahui fungsionalitas QuickASP secara umum.

5.2.1 Skenario Pertama

Tujuan skenario yang pertama adalah untuk menguji QuickASP dalam proses pembangkitan kode ASP dengan masukan sebagai berikut :

- Tipe basis data : SQL Server
- Nama basis data : Northwind
- Nama tabel : Employees
- File CSS : Ocean.css
- Metode pembangkitan : Manual

Pada uji coba pertama ini, *server* yang digunakan sebagai tempat menyimpan file-file ASP hasil pembangkitan kode ASPWebGen merupakan komputer lokal.



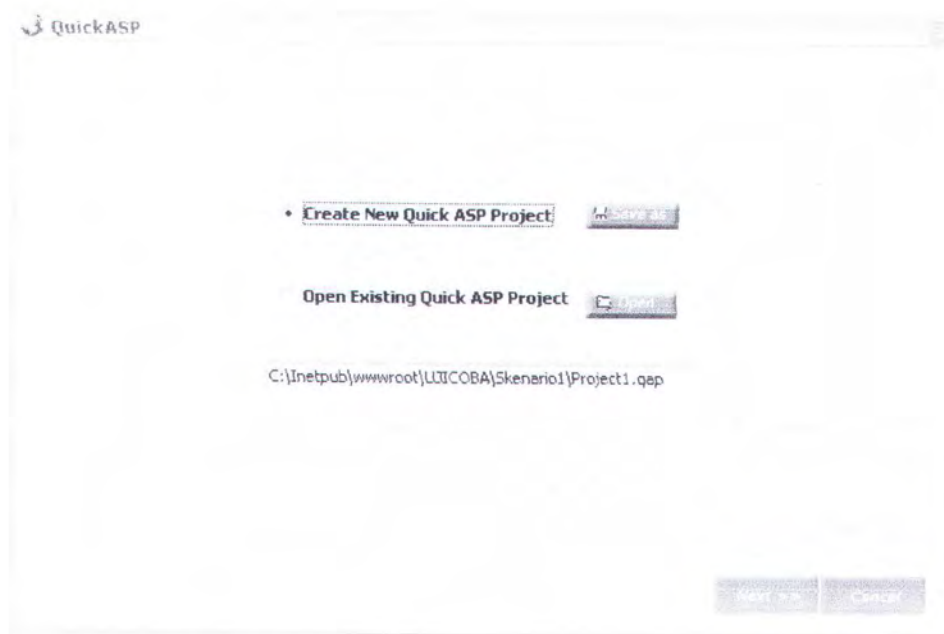
Gambar 5.1 Struktur Basis Data *Northwind* pada MS SQL Server

▪ Langkah – Langkah Uji Coba

Langkah-langkah uji coba skenario pertama adalah sebagai berikut :

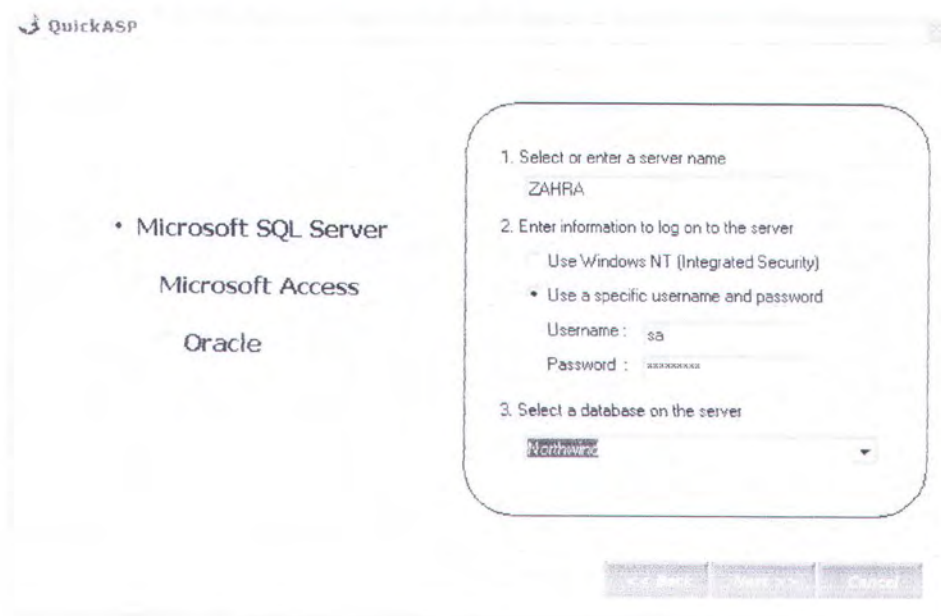
1. Membuat project baru yaitu Project1.qap yang disimpan pada direktori

C:\Inetpub\wwwroot\UJICOBA\Skenario1\



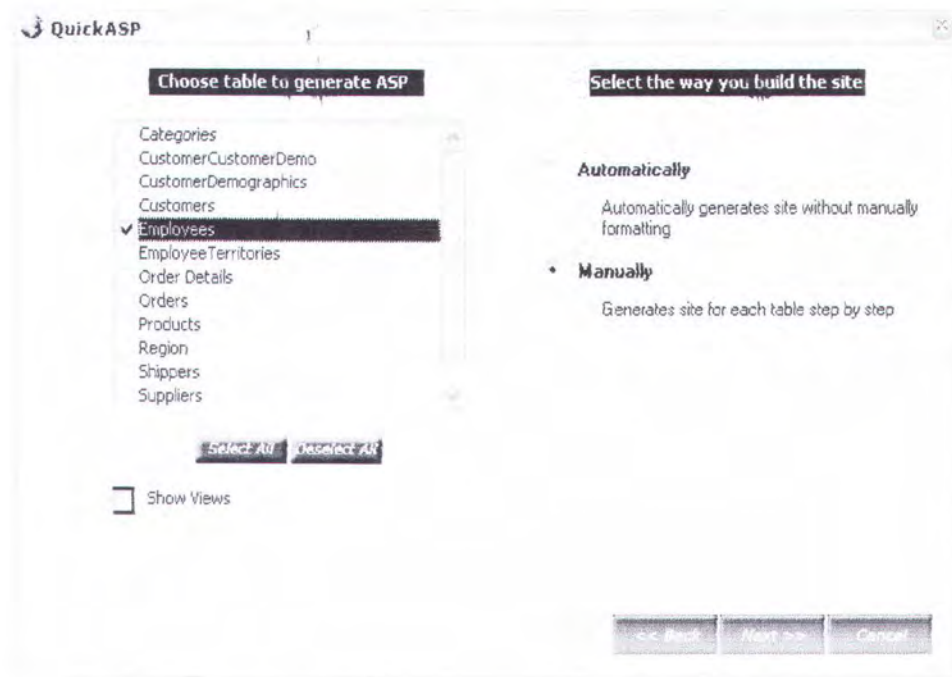
Gambar 5.2 Form Pemilihan Proyek (Skenario 1)

2. Memilih tipe basis data Microsoft SQL Server dan memasukkan informasi basis data yang hendak dibuka yaitu basis data "Northwind"



Gambar 5.3 Form Pemilihan Basis Data

3. Memilih tabel yang hendak dibangkitkan yaitu tabel “Employees” dan juga memilih metode pembangkitan yaitu *Manually*.



Gambar 5.4 Form Pemilihan Tabel dan Metode Pembangkitan (Skenario 1)

4. Memilih *field* yang hendak ditampilkan pada halaman daftar. Pada skenario ini hanya *field* “EmployeeID”, “FirstName”, dan “LastName” yang hendak ditampilkan pada halaman daftar.

Quick ASP

Shortcut

Project Option

DB Connection

User Guide

About Me

:: Employees ::

Select fields that will appear in list page

EmployeeID	EmployeeID
TitleOfCourtesy	FirstName
BirthDate	LastName
HireDate	
Address	
City	
Region	
PostalCode	
Country	

Order data by field: EmployeeID Descending

Number of records on each page: 10

☐ Show first n characters of large fields on the list page
 'More ...' link displays the whole field

GO

Back Next Cancel

Gambar 5.5 Form Pemilihan *Field* Halaman Daftar (Skenario 1)

- Melakukan pengaturan halaman daftar. Setiap kali pengguna memasuki form ini, secara *default* aplikasi telah menyediakan format tampilan halaman web yang akan dihasilkan sebagaimana tampak pada gambar 5.6. Pada skenario ini pengguna mengubah tipe tampilan untuk *field* "HireDate" yang semula "ShortDate[Time]" menjadi "LongDate[Time]". Selain itu pengguna juga mengubah tipe tampilan untuk *field* "PhotoPath" yang semula "Text" menjadi "File-based Image". Hal ini dimaksudkan agar nantinya halaman web yang dihasilkan mampu menampilkan data yang tersimpan pada basis data dalam bentuk gambar. Hasil dari pengubahan format tampilan tersebut dapat dilihat pada gambar 5.7.

QuickASP

Shortcut

:: Employees ::

Project Option

DB Connection

User Guide

About Me

Select fields that will appear in list page

TitleOfCourtesy	EmployeeID
BirthDate	FirstName
HireDate	LastName
Address	
City	
Region	
PostalCode	
Country	

Order data by field : EmployeeID Descending

Number of records on each page : 10

☐ Show first n characters of large fields on the list page
"More ..." link displays the whole field

Gambar 5.5 Form Pemilihan *Field* Halaman Daftar (Skenario 1)

- Melakukan pengaturan halaman daftar. Setiap kali pengguna memasuki form ini, secara *default* aplikasi telah menyediakan format tampilan halaman web yang akan dihasilkan sebagaimana tampak pada gambar 5.6. Pada skenario ini pengguna mengubah tipe tampilan untuk *field* "HireDate" yang semula "ShortDate[Time]" menjadi "LongDate[Time]". Selain itu pengguna juga mengubah tipe tampilan untuk *field* "PhotoPath" yang semula "Text" menjadi "File-based Image". Hal ini dimaksudkan agar nantinya halaman web yang dihasilkan mampu menampilkan data yang tersimpan pada basis data dalam bentuk gambar. Hasil dari pengubahan format tampilan tersebut dapat dilihat pada gambar 5.7.

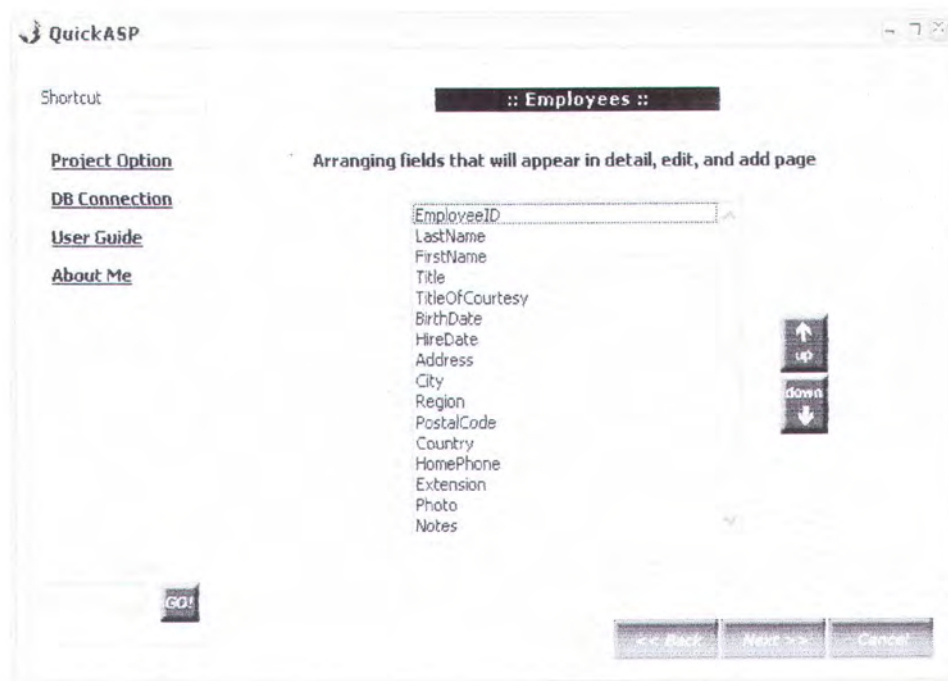
Field	Label	List Page	Detail Page	View as	View Format
EmployeeID	EmployeeID			Numeric	Width=60; Align=Left
LastName	LastName			Text	Width=200; Align=Left
FirstName	FirstName			Text	Width=200; Align=Left
Title	Title	X		Text	Width=200; Align=Left
TitleOfCourtesy	TitleOfCourtesy	X		Text	Width=200; Align=Left
BirthDate	BirthDate	X		ShortDate[Time]	Width=50; Align=Left; Format=dd/mm/yyyy;
HireDate	HireDate	X		ShortDate[Time]	Width=50; Align=Left; Format=dd/mm/yyyy;
Address	Address	X		Text	Width=200; Align=Left
City	City	X		Text	Width=200; Align=Left
Region	Region	X		Text	Width=200; Align=Left
PostalCode	PostalCode	X		Text	Width=200; Align=Left
Country	Country	X		Text	Width=200; Align=Left
HomePhone	HomePhone	X		Text	Width=200; Align=Left
Extension	Extension	X		Text	Width=200; Align=Left
Photo	Photo	X		Database Image	Width=60; ImageWidth=60; ImageHeight=60
Notes	Notes	X		Text	Width=200; Align=Left
ReportsTo	ReportsTo	X		Text	Width=60; Align=Left; TableReference=Employees;
PhotoPath	PhotoPath	X		Text	Width=200; Align=Left

Gambar 5.6 Form Pengaturan Halaman Daftar (*default*)

Field	View as	View Format
BirthDate	ShortDate[Time]	Width=50; Align=Left; Format=dd/mm/yyyy;
HireDate	LongDate[Time]	Width=300; Align=Left; Format=mm/dd/yyyy;
Address	Text	Width=200; Align=Left
City	Text	Width=200; Align=Left
Region	Text	Width=200; Align=Left
PostalCode	Text	Width=200; Align=Left
Country	Text	Width=200; Align=Left
HomePhone	Text	Width=200; Align=Left
Extension	Text	Width=200; Align=Left
Photo	Database Image	Width=60; ImageWidth=60; ImageHeight=60
Notes	Text	Width=200; Align=Left
ReportsTo	Text	Width=60; Align=Left; TableReference=Employees;
PhotoPath	File-based Image	Width=60; ImageWidth=60; ImageHeight=60

Gambar 5.7 Form Pengaturan Halaman Daftar (setelah diubah)

6. Melakukan pengaturan urutan *field* pada halaman ubah. Pada skenario ini, pengguna tidak melakukan pengubahan urutan *field* yang hendak ditampilkan pada halaman ubah (*Edit Page*).



Gambar 5.8 Form Pengaturan Susunan *Field* pada Halaman Ubah (Skenario 1)

7. Melakukan pengaturan tampilan masukan pada halaman ubah. Setiap kali pengguna memasuki form ini, secara *default* aplikasi telah menyediakan format tampilan halaman web yang akan dihasilkan sebagaimana tampak pada gambar 5.9.

QuickASP

Shortcut

:: Employees ::

Project Option

DB Connection

User Guide

About Me

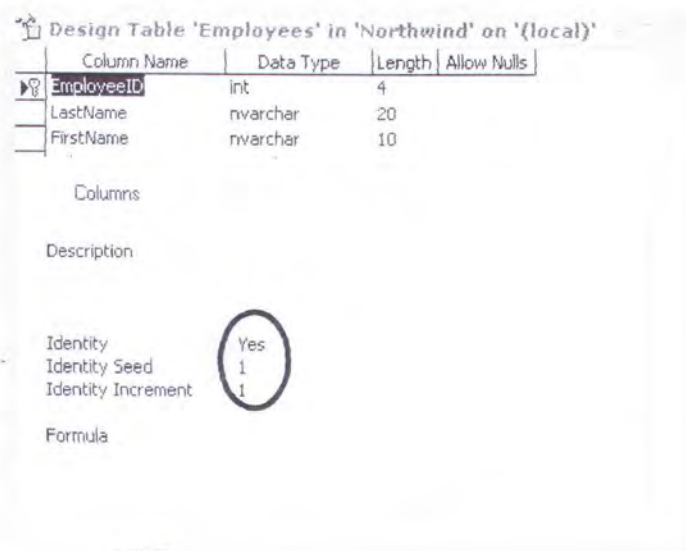
Customize your detail and edit page appearance, click the cell to edit it

Field	Label	Edit	Add	Edit as	
EmployeeID	EmployeeID	X	X		
LastName	LastName			Text Field	Size=60; MaxLength=20; Tipe=
FirstName	FirstName			Text Field	Size=60; MaxLength=10; Tipe=
Title	Title			Text Field	Size=60; MaxLength=30; Tipe=
TitleOfCourtesy	TitleOfCourtesy			Text Field	Size=60; MaxLength=25; Tipe=
BirthDate	BirthDate			Date	Style=Textbox; Format=dd/mm
HireDate	HireDate			Date	Style=Textbox; Format=dd/mm
Address	Address			Text Field	Size=60; MaxLength=60; Tipe=
City	City			Text Field	Size=60; MaxLength=15; Tipe=
Region	Region			Text Field	Size=60; MaxLength=15; Tipe=
PostalCode	PostalCode			Text Field	Size=60; MaxLength=10; Tipe=
Country	Country			Text Field	Size=60; MaxLength=15; Tipe=
HomePhone	HomePhone			Text Field	Size=60; MaxLength=24; Tipe=

OK Cancel

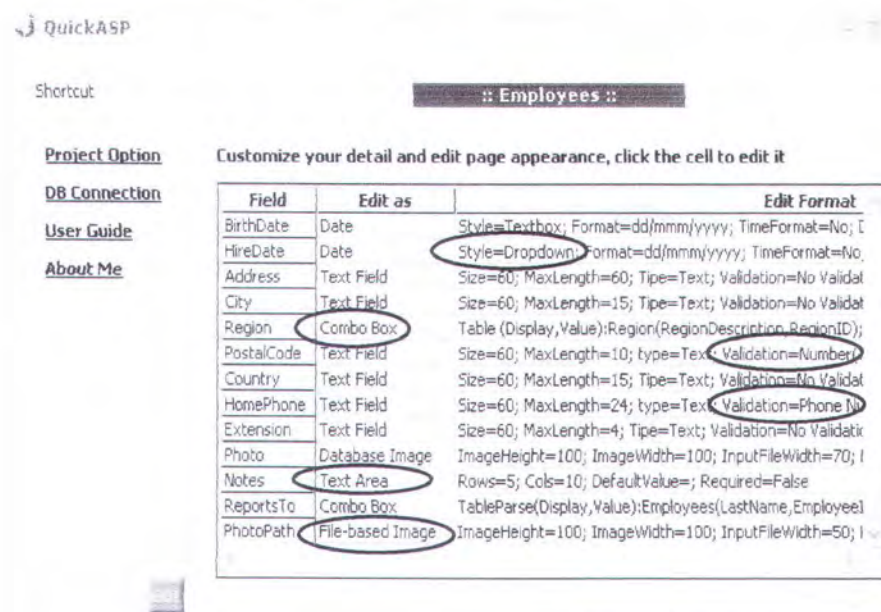
Gambar 5.9 Form Pengaturan Halaman Ubah Skenario 1 (default)

Terlihat bahwa pada *field* "EmployeeID", secara otomatis kolom Edit dan kolom Add disilang yang berarti bahwa nantinya pengunjung web tidak akan bisa mengubah maupun menambah data baru untuk *field* tersebut. Hal ini dikarenakan *field* tersebut termasuk dalam kategori *Identity Field* (*autoinsert*) sebagaimana yang terlihat pada gambar desain basis data di bawah ini.



Gambar 5.10 Desain Basis Data “Northwind” Tabel “Employees”

Pada skenario ini pengguna mengubah beberapa tampilan masukan sebagaimana yang terlihat pada gambar di bawah:



Gambar 5.11 Form Pengaturan Halaman Ubah Skenario 1 (setelah diubah)

Gambar 5.15 Pemilihan Direktori Keluaran dan URL Website (Skenario 1)

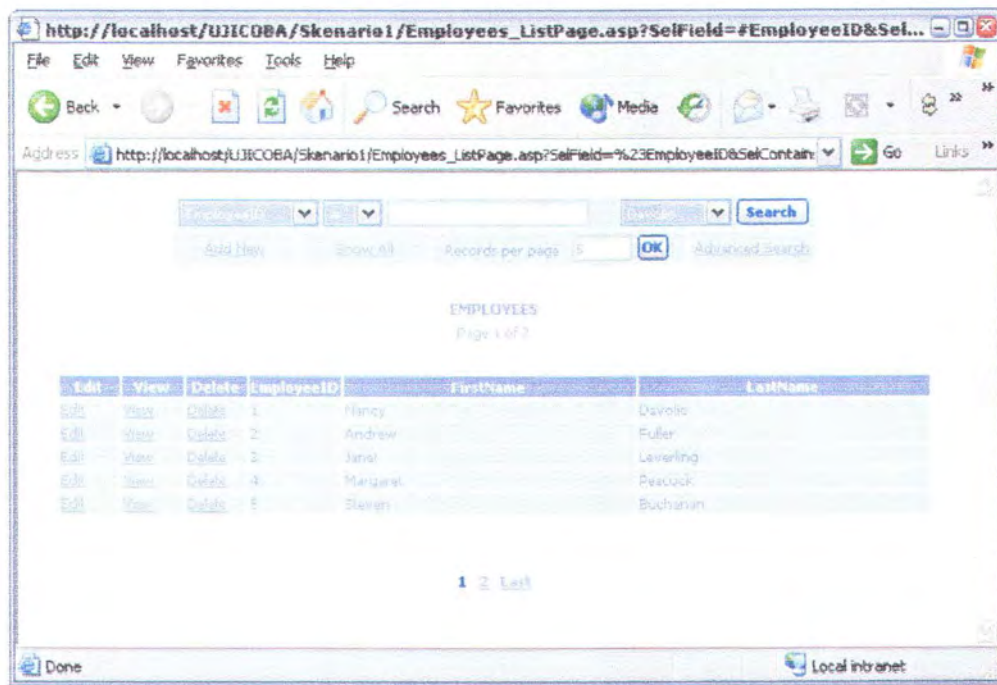
11. Selanjutnya pengguna tinggal menekan tombol “Build” untuk membangkitkan file-file ASP dan file-file pendukung lainnya.

File	Directory Path
...	C:\inetpub\wwwroot\UJICOBA\Skenario
...	C:\inetpub\wwwroot\UJICOBA\Skenario
...	Date selection ASP Function
...	Date selection JavaScript function
...	C:\inetpub\wwwroot\UJICOBA\Skenario
...	C:\inetpub\wwwroot\UJICOBA\Skenario
...	C:\inetpub\wwwroot\UJICOBA\Skenario
...	C:\inetpub\wwwroot\UJICOBA\Skenario
...	Edit-update record page

Gambar 5.16 Form Pembangkitan Kode Program (Skenario 1)

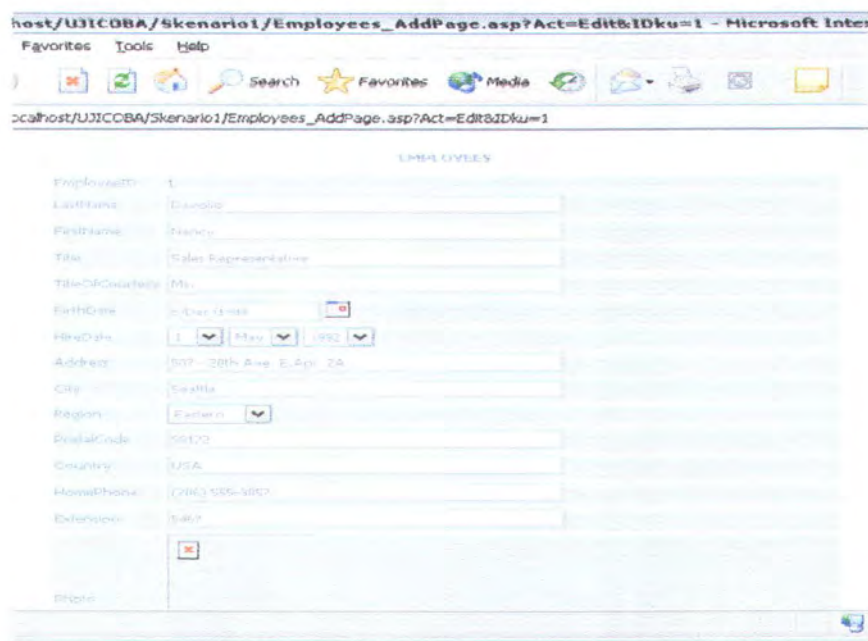
▪ **Tampilan Halaman Web Hasil Pembangkitan**

Halaman Daftar (Employees ListPage.asp)



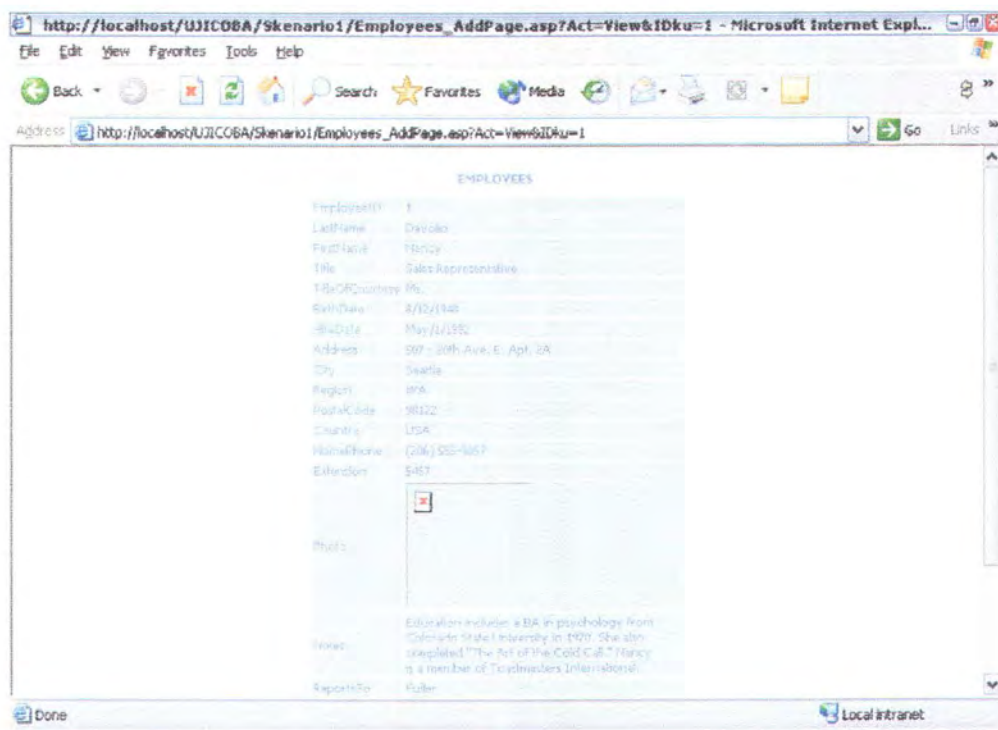
Gambar 5.17 Tampilan Halaman Daftar (Skenario 1)

Halaman Ubah (Employees_AddPage.asp?Act=Edit)



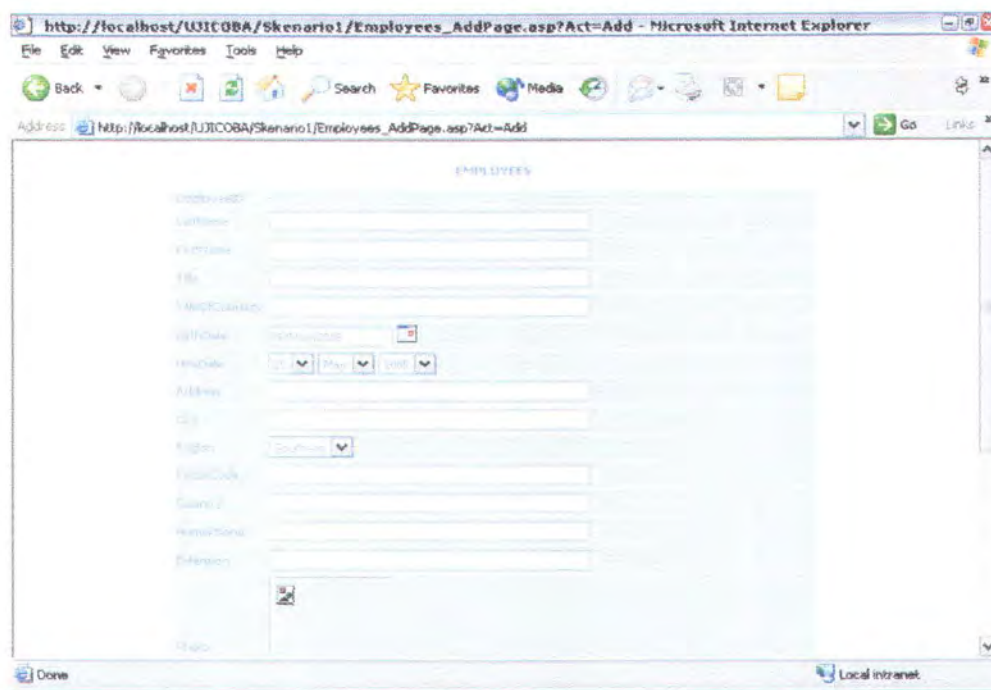
Gambar 5.18 Tampilan Halaman Ubah (Skenario 1)

Halaman Tampil (Employees AddPage.asp?Act=View)



Gambar 5.19 Tampilan Halaman Tampil (Skenario 1)

Halaman Tambah (Employees AddPage.asp?Act=Add)



Gambar 5.20 Tampilan Halaman Tambah (Skenario 1)

▪ Evaluasi Hasil Pembangunan

Dari ketiga halaman hasil pembangunan (*List Page*, *Detail Page*, *Edit Page*) terlihat bahwa *field* “Photo” tidak dapat menampilkan gambar yang dimaksud karena format gambar yang tersimpan pada tabel “Employees” tidak dikenali oleh aplikasi. Aplikasi QuickASP hanya mampu menampilkan gambar dalam format jpeg, gif, dan bmp.

5.2.2 Skenario Kedua

Pada skenario yang ke dua, data masukannya sebagai berikut:

- Tipe basis data : SQL Server
- Nama basis data : pubs

- Nama tabel : authors, discounts, employee, jobs, pubinfo, publishers, roysched, sales, stores, titleauthor, titles
- File CSS : Nature.css
- Metode pembangkitan : Automatically

Pada uji coba kedua ini, *server* yang digunakan sebagai tempat menyimpan file-file ASP hasil pembangkitan kode ASPWebGen merupakan komputer lokal. Perbedaan utama antara skenario ke dua dengan skenario pertama adalah metode pembangkitan yang digunakan. Pada skenario kedua ini, pengguna tidak perlu melakukan pengaturan satu per satu untuk tiap tabel yang hendak dibangkitkan ke dalam kode ASP.

Berikut gambar pengaturan metode pembangkitan pada skenario kedua ini:

The screenshot shows the QuickASP web application interface. It has a title bar with a logo and the text "QuickASP". The main content area is divided into two panels. The left panel, titled "Choose table to generate ASP", contains a list of tables with checkboxes next to them: authors, discounts, employee, jobs, pub_info, publishers, roysched, sales, stores, titleauthor, and titles. All checkboxes are checked. Below the list are two buttons: "Select All" and "Deselect All". At the bottom of this panel is a checkbox labeled "Show Views". The right panel, titled "Select the way you build the site", contains two radio button options: "Automatically" (selected) and "Manually". The "Automatically" option has a description: "Automatically generates site without manually formatting". The "Manually" option has a description: "Generates site for each table step by step". At the bottom of the right panel are three buttons: "Build", "Cancel", and "Help".

Gambar 5.21 Form Pemilihan Tabel dan Metode Pembangkitan (Skenario 2)

Berikut file-file hasil pembangkitan yang dilakukan oleh QuickASP pada skenario kedua:

```

include
[?] authors_AddPage.asp
[?] authors_ListPage.asp
[?] authors_SearchPage.asp
[?] discounts_AddPage.asp
[?] discounts_ListPage.asp
[?] discounts_SearchPage.asp
[?] employee_AddPage.asp
[?] employee_ListPage.asp
[?] employee_SearchPage.asp
[?] jobs_AddPage.asp
[?] jobs_ListPage.asp
[?] jobs_SearchPage.asp
[?] pub_info_AddPage.asp
[?] pub_info_ListPage.asp
[?] pub_info_SearchPage.asp
[?] publishers_AddPage.asp
[?] publishers_ListPage.asp
[?] publishers_SearchPage.asp
[?] roysched_AddPage.asp
[?] roysched_ListPage.asp
[?] roysched_SearchPage.asp
[?] sales_AddPage.asp
[?] sales_ListPage.asp
[?] sales_SearchPage.asp
[?] stores_AddPage.asp
[?] stores_ListPage.asp
[?] stores_SearchPage.asp
[?] titleauthor_AddPage.asp
[?] titleauthor_ListPage.asp
[?] titleauthor_SearchPage.asp
[?] titles_AddPage.asp
[?] titles_ListPage.asp
[?] titles_SearchPage.asp
[?] Project2.qap

```

Gambar 5.22 File-File Hasil Pembangkitan Kode pada Skenario Kedua

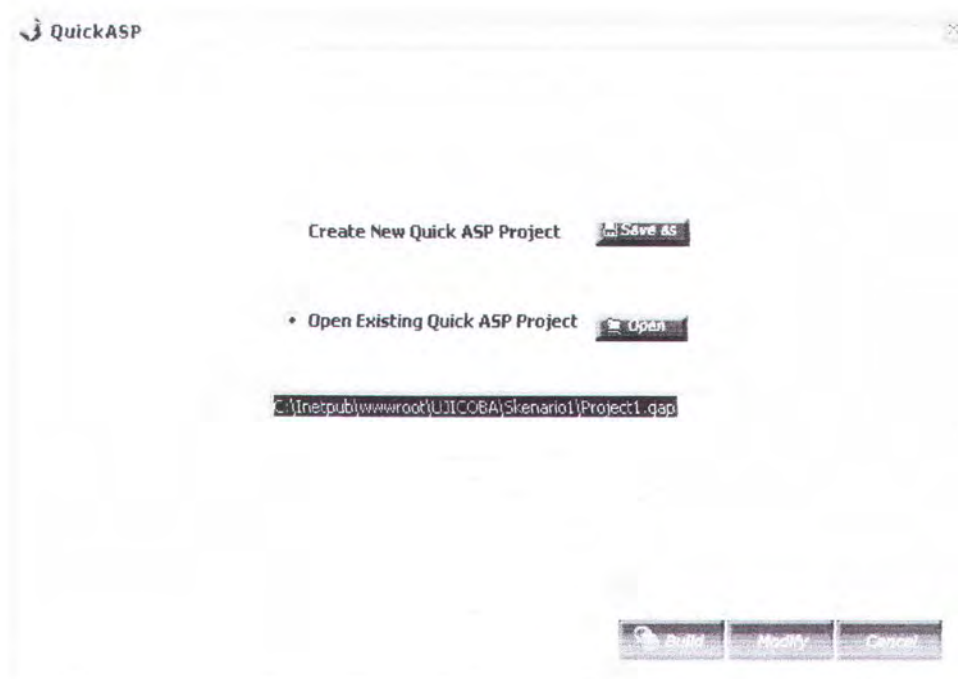
5.2.3 Skenario Ketiga

Pada skenario ketiga, aplikasi akan membuka file proyek yang sudah dibuat sebelumnya lalu melakukan modifikasi.

▪ Langkah – Langkah Uji Coba

1. Membuka file proyek yang sudah ada, yaitu

C:\inetpub\wwwroot\UJICOBA\Skenario1\Project1.qap



Gambar 5.23 Form Pemilihan Proyek yang Sudah Ada (Skenario 3)

2. Setiap kali pengguna membuka form pengaturan tampilan, maka secara otomatis aplikasi QuickASP akan menampilkan hasil pengaturan pada proyek yang dibuka. Misal pada gambar 5.24 terlihat bahwa *field* yang ditampilkan pada halaman daftar yaitu *field* "EmployeeID", "FirstName", dan "LastName". Hal ini sama dengan pengaturan pada skenario pertama.



QuickASP

Shortcut

:: Employees ::

Project Option

DB Connection

User Guide

About Me

Select fields that will appear in list page

Title	EmployeeID
TitleOfCourtesy	FirstName
BirthDate	LastName
HireDate	
Address	
City	
Region	
PostalCode	
Country	

Order data by field: EmployeeID Descending

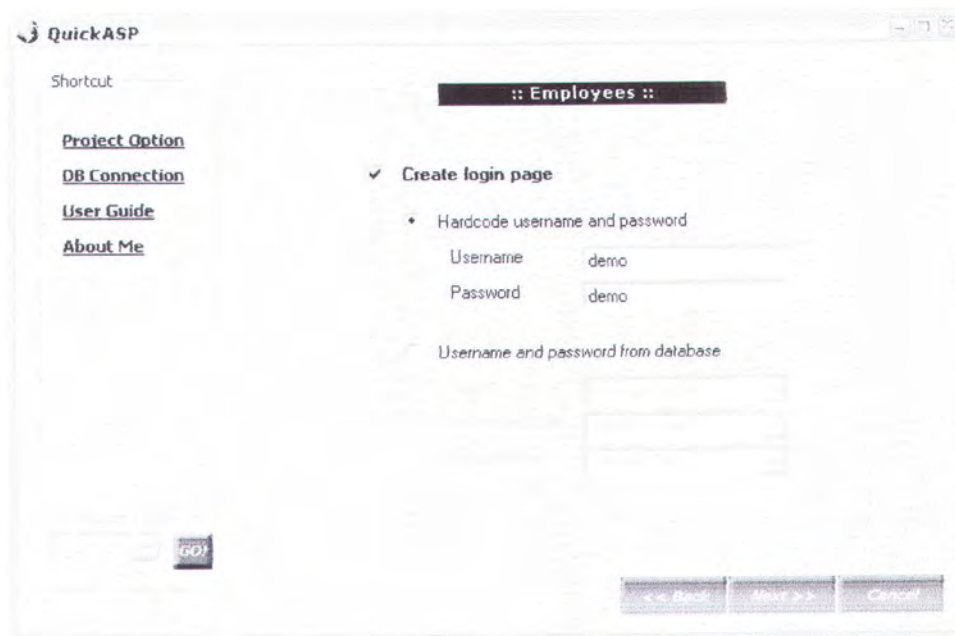
Number of records on each page: 10

☒ Show first n characters of large fields on the list page
 "More ..." link displays the whole field Character number to display:

Back Next Cancel

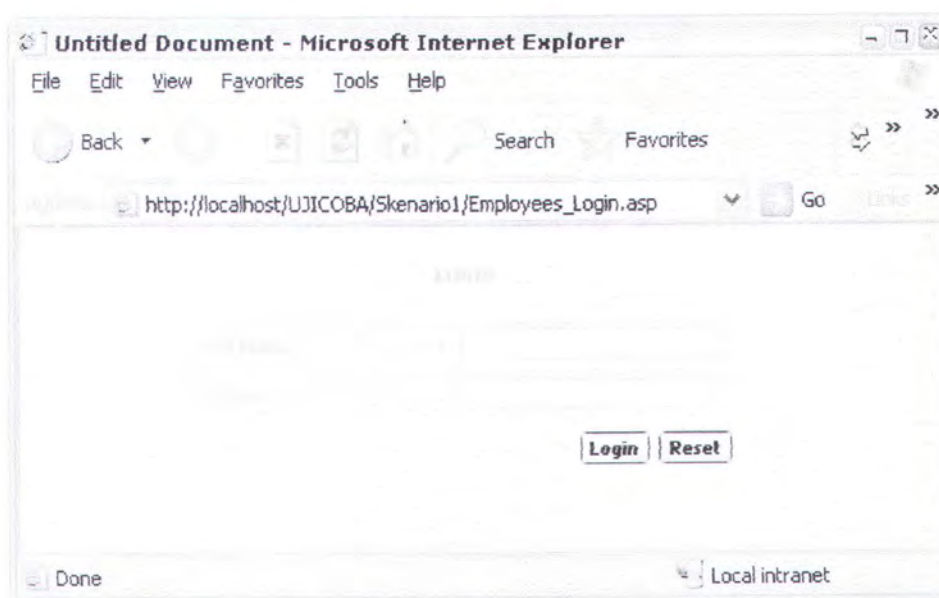
Gambar 5.24 Form Pengaturan Halaman Daftar pada Proyek yang Dibuka
(Skenario 3)

3. Melakukan modifikasi yaitu menambahkan halaman login.
4. Melakukan pembangkitan kode program. Aplikasi akan menanyakan kepada pengguna apakah pengguna ingin mengganti file yang sudah ada dengan file yang baru terbentuk. Jika ya, maka aplikasi akan mengganti file lama dengan file baru.



Gambar 5.25 Form Pengaturan Halaman Login (Skenario 3)

▪ **Tampilan Halaman Hasil Pembangkitan**



Gambar 5.26 Tampilan Halaman Login (Skenario 3)

5.2.4 Skenario Keempat

Pada skenario keempat ini, aplikasi melakukan pembangkitan kode program ASP dengan masukan sebagai berikut :

- Tipe basis data : Oracle
- Nama basis data : RAHMI
- Nama tabel : Anggota
- File CSS : Rose.css
- Metode pembangkitan : Manual

Selain itu pada skenario ini akan dilakukan pembangkitan halaman login dengan berbagai pilihan tingkat keamanan.

▪ Langkah – Langkah Uji Coba

1. Membuat project baru yaitu Project1.qap yang disimpan pada direktori C:\inetpub\wwwroot\UJICOBA\Skenario4\
2. Memasukkan properti basis data Oracle yang meliputi nama server, *user id*, dan *password*.
3. Memilih tabel yang hendak dibangkitkan yaitu tabel “ANGGOTA”.

Column	Pk	Data Type	Null?
IDANGGOTA	1	NUMBER (10)	N
NAMA		VARCHAR2 (30)	N
IDJENISANGGOTA		NUMBER	Y
ALAMAT		VARCHAR2 (50)	N
JK		NUMBER (2)	N
TGLLAHIR		DATE	Y
EMAIL		VARCHAR2 (50)	Y
USERID		NUMBER	Y

Gambar 5.27 Properti Kolom dari Tabel “ANGGOTA” (Skenario 4)

4. Memilih kolom yang hendak ditampilkan pada halaman daftar. Pada skenario ini, kolom yang dipilih yaitu: IDANGGOTA, NAMA, ALAMAT. Selain itu agar tabel pada halaman daftar tidak terlalu penuh, maka jumlah karakter yang ditampilkan dibatasi maksimal 30 karakter. Jika panjang karakter melebihi batasan tersebut maka *link* "Detail" akan muncul untuk menampilkan isi keseluruhan.
5. Melakukan pengaturan format tampilan halaman daftar. Pada bagian ini tidak dilakukan perubahan format tampilan.
6. Melakukan pengaturan format tampilan halaman ubah.

Field	Label	Edit	Add	Edit as
IDANGGOTA	IDANGGOTA			Text Field
NAMA	NAMA			Text Field
IDJENISANGGOTA	IDJENISANGGOTA			Combo Box
ALAMAT	ALAMAT			Text Field
JK	JK			Text Field
TGLLAHIR	TGLLAHIR			Date
EMAIL	EMAIL			Text Field
USERID	USERID	X	X	

Edit Format

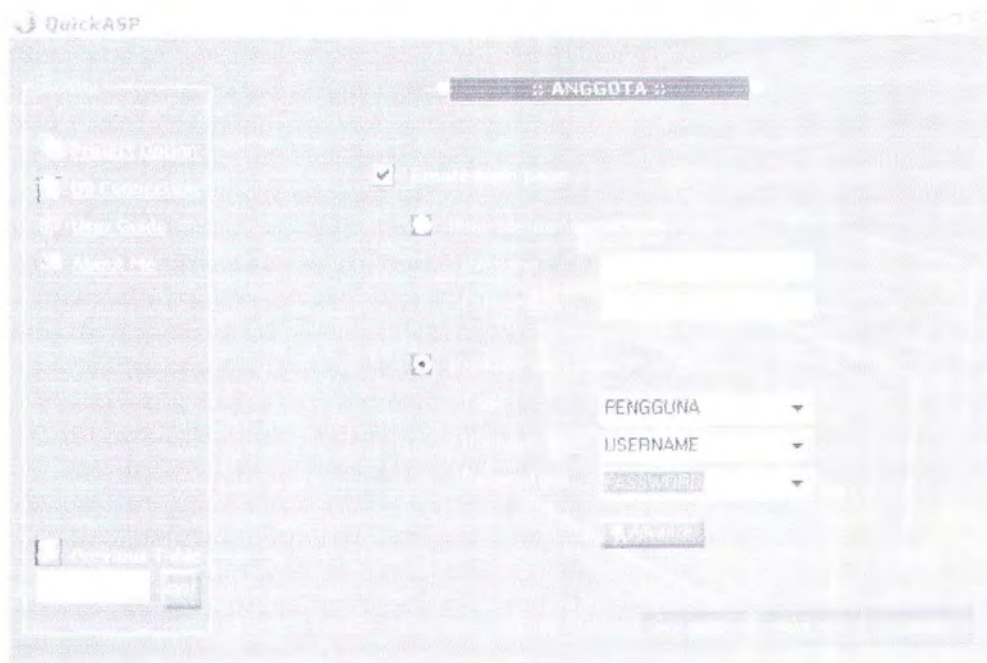
Size=40; MaxLength=22; Tipe=Text; Validation=Number(###); DefaultValue=; Required=Fals
 Size=120; MaxLength=30; Tipe=Text; Validation=No Validation; DefaultValue=; Required=Fals
 TableParse(Display, Value):JENISANGGOTA(JENISANGGOTA, IDJENISANGGOTA); Required=Fals
 Size=120; MaxLength=50; Tipe=Text; Validation=No Validation; DefaultValue=; Required=Fals
 Size=40; MaxLength=22; Tipe=Text; Validation=Number(###); DefaultValue=; Required=Fals
 Style=Textbox; Format=dd/mm/yyyy; TimeFormat=No; Datepicker=Yes; YearRange=No;
 Size=120; MaxLength=50; Tipe=Text; Validation=No Validation; DefaultValue=; Required=Fals

Gambar 5.28 Pengaturan Format Tampilan Halaman Ubah (Skenario 4)

Sebagaimana yang terlihat pada gambar di atas, kolom Edit dan Add pada *field* "USERID" diberi tanda silang sehingga nantinya pada halaman web yang dihasilkan pengunjung web tidak dapat menambah USERID baru

maupun mengubahnya. Karena perubahan pada tabel PENGGUNA hanya dapat dilakukan oleh administrator.

7. Melakukan pengaturan halaman pencarian
8. Melakukan pengaturan halaman login. *Username* dan *password* diambil dari tabel "PENGGUNA" yaitu kolom "USERNAME" dan "PASSWORD":



Gambar 5.29 Pengaturan Format Tampilan Halaman Login (Skenario 4)

9. Agar tingkat keamanan pada halaman login lebih tinggi, tekan tombol "Advance" sehingga muncul form berikut:



Gambar 5.30 Pengaturan Format Tampilan Halaman Login Lanjut (Skenario 4)

Pada skenario ini dipilih jenis sekuritas ke-2 dimana pengunjung web hanya dapat melihat, mengubah, dan menghapus data miliknya. Kolom "USERID" dari tabel "PENGGUNA" dipasangkan dengan kolom "USERID" dari tabel "ANGGOTA". Selain itu *username* "aaa" dipilih sebagai *username* yang berfungsi sebagai administrator. Administrator dapat melakukan penambahan, pengubahan dan penghapusan seluruh data pada tabel. Sedangkan "Login Guest" hanya dapat melihat data pada halaman daftar tanpa melakukan penambahan data baru maupun pengubahan data yang sudah ada.

10. Memilih *stylesheet* dan direktori keluaran.

11. Melakukan pembangkitan kode program.

▪ Tampilan Halaman Hasil Pembangkitan

1) Login sebagai pengguna biasa

Pada gambar 5.31 username “ika” memasukkan login dan menuju pada halaman daftar (gambar 5.32).

ANGGOTA LOGIN	
User Name	ika
Password	*****
<input type="button" value="Login"/> <input type="button" value="Reset"/>	

Gambar 5.31 Tampilan Halaman Login

Logged as ika

Records per page: 10 Quick Search: AnyField GO!

Page 1 of 1

Edit	View	Delete	IDANGGOTA	NAMA	ALAMAT
			1	rahmi	Jl Wisma Menanggal 1/5 Surabaya ...
			2	ika	Jl Manyar Kartika 56 Bandung, ...
			3	tanto	Jl Untung Suropati No 35 Pasur ...

Gambar 5.32 Tampilan Halaman Daftar

Pada gambar 5.31 username “ika” memasukkan login dan menuju pada halaman daftar (gambar 5.32).

ANGGOTA LOGIN

User Name: ika

Password:

Login Reset

[Back to menu](#) [Login as Guest](#)

Gambar 5.31 Tampilan Halaman Login

Logged as ika [Logout](#) [Back to menu](#) [Advanced search](#) [Add new](#)

Show All Records per page: 10 Quick Search: AnyField [Go!](#)

Page 1 of 1

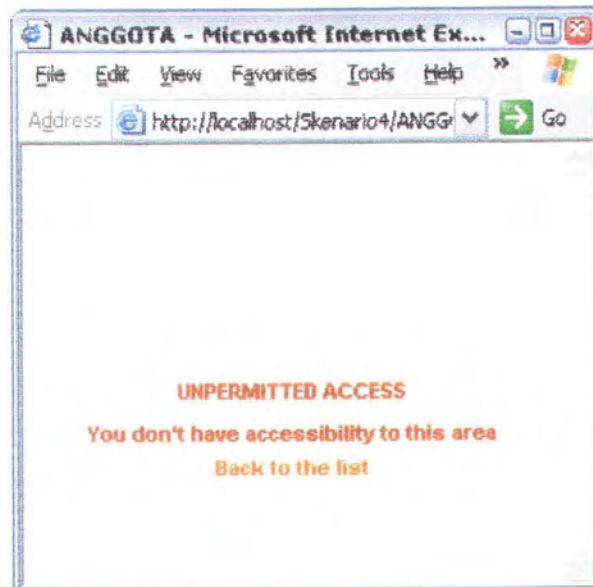
ANGGOTA

Edit	View	Delete	IDANGGOTA	NAMA	ALAMAT
Edit	View	Delete	1	rahmi	Jl Yloma Merianggal 15 Surabaya ... Detail
Edit	View	Delete	2	ika	Jl Menyer Kartika 56 Bandung, ... Detail
Edit	View	Delete	3	tanto	Jl Lintang Suropati No 35 Pasur ... Detail

1

Gambar 5.32 Tampilan Halaman Daftar

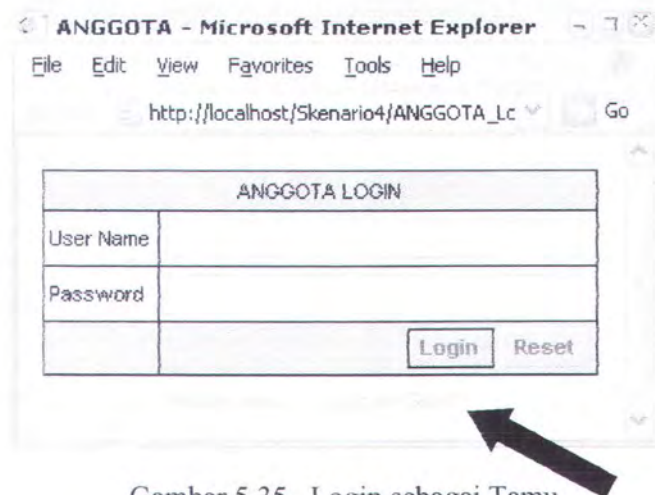
Ketika *username* “ika” mencoba mengubah data pengunjung lain, maka keluar pesan error sebagai berikut:



Gambar 5.33 Pesan Error

Namun ketika *username* “ika” hendak mengubah data pribadinya maka muncul halaman berikut:

Gambar 5.34 Tampilan Halaman Ubah

2) *Guest Login*

Gambar 5.35 Login sebagai Tamu



Gambar 5.36 Halaman Daftar untuk Login sebagai Tamu

Pada gambar 5.36 tidak terlihat *link* "Edit", "Delete", "View" maupun "Add New".

BAB VI

PENUTUP

Setelah dilakukan pembahasan teoritis dan praktis pada bab-bab sebelumnya dan pengujian QuickASP beserta analisisnya, maka pada bab penutup ini akan diambil kesimpulan serta kemungkinan pengembangan dari Tugas Akhir ini.

6.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil dari proses pembuatan QuickASP untuk Tugas Akhir ini adalah:

1. QuickASP dapat melakukan proses pembangkitan kode ASP dari basis data Microsoft Access, Microsoft SQL Server, dan Oracle.
2. QuickASP dapat menghasilkan homepage dinamis. Homepage yang dihasilkan dilengkapi dengan fungsi menambah data, mengubah data, menghapus data dan menu navigasi.
3. QuickASP dapat menampilkan pada halaman web data-data dengan tipe data yaitu *text*, *numeric*, *date*, *boolean* dan *image* (.gif, .jpg, .bmp).
4. QuickASP merupakan aplikasi desktop, bukan aplikasi berbasis web. Sehingga untuk menggunakan QuickASP, pengguna perlu melakukan instalasi QuickASP di komputer lokal.

6.2 Kemungkinan Pengembangan

Beberapa saran yang dapat penulis berikan untuk Tugas Akhir ini adalah :

1. QuickASP saat ini berupa aplikasi desktop, sehingga bisa dikembangkan menjadi aplikasi QuickASP berbasis web.
2. Halaman yang dihasilkan bisa dikembangkan ke bahasa pemrograman web lainnya seperti JSP dan PHP sehingga memberikan banyak pilihan bagi pihak pengguna aplikasi.
3. Adanya fitur pemilihan *template* sehingga pengguna dapat memiliki banyak pilihan desain web yang beragam dan menarik.
4. Penambahan jenis DBMS yang dapat diakses oleh QuickASP dan homepage yang dibangkitkan, seperti MySQL.

DAFTAR PUSTAKA

- [1] ASP tutorial at <http://w3schools.com/asp>, 1999.
- [2] Castagnetto, Jesus, and Rawat, Harish, and Schumann, Sascha, *Professional PHP Programing*, Chris Scollo, Deepak Veliath, 1999.
- [3] Isharyono, Andrie, *P3L Database Management Berbasis Web untuk Database MySQL dengan Menggunakan PHP*, 2001.
- [4] Kurniawan, Yahya, S.T, *Aplikasi Web Database dengan ASP*, PT.Elex Media Komputindo, Jakarta, 2002.
- [5] Microsoft Corporation, Microsoft Developer Network Library, <http://msdn.microsoft.com> Juli 2001.
- [6] Penerbit ANDI, *Panduan Aplikatif Pengembangan Web Berbasis ASP*, Yogyakarta, 2002

LAMPIRAN

a. Tabel View Information_Schema SQL Server

INFORMATION_SCHEMA VIEW	COLUMN NAME
Information_Schema.Columns	<ul style="list-style-type: none"> - table_catalog - table_schema - table_name - column_name - ordinal_position - column_default - is_nullable - data_type - character_maximum_length - character_octet_length - numeric_precision - numeric_precision_radix - numeric_scale - datetime_precision - character_set_catalog - character_set_schema - character_set_name - collation_catalog - collation_schema - collation_name - domain_catalog - domain_schema - domain_name
Information_Schema.Key_Column_Usage	<ul style="list-style-type: none"> - constraint_catalog - constraint_schema - constraint_name - table_catalog - table_schema - table_name - column_name - ordinal_position
Information_Schema.Referential_Constraints	<ul style="list-style-type: none"> - constraint_catalog - constraint_schema - constraint_name - unique_constraint_catalog - unique_constraint_schema - unique_constraint_name - match_option

	<ul style="list-style-type: none"> - update_rule - delete_rule
Information_Schema.Schemata	<ul style="list-style-type: none"> - catalog_name - schema_name - schema_owner - default_character_set_catalog - default_character_set_schema - default_character_set_name
Information_Schema.Tables	<ul style="list-style-type: none"> - table_catalog - table_schema - table_name - table_type
Information_Schema.Views	<ul style="list-style-type: none"> - table_catalog - table_schema - table_name - view_definition - check_option - is_updatable

b. Tabel ColumnProperty() SQL Server

Nilai	Deskripsi	Nilai Balik
AllowsNull	Untuk mengecek apakah kolom tersebut <i>Nullable</i> atau bukan	1 = TRUE 0 = FALSE NULL = Invalid input
IsComputed	Untuk mengecek apakah kolom tersebut termasuk <i>Computed Column</i>	1 = TRUE 0 = FALSE NULL = Invalid input
IsCursorType	Untuk mengecek apakah parameter prosedur tersebut termasuk <i>Cursor</i> atau bukan	1 = TRUE 0 = FALSE NULL = Invalid input
IsDeterministic	Untuk mengecek apakah kolom tersebut <i>deterministic</i> . Properti ini hanya bisa diterapkan pada <i>computed columns</i> dan <i>view columns</i> .	1 = TRUE 0 = FALSE NULL = Invalid input. Not a computed column or view column.
IsFulltextIndexed	Untuk mengecek apakah kolom tersebut <i>full-text indexing</i> .	1 = TRUE 0 = FALSE NULL = Invalid input

IsIdentity	Untuk mengecek apakah kolom tersebut IDENTITY .	1 = TRUE 0 = FALSE NULL = Invalid input
IsIdNotForRepl	Untuk mengecek apakah kolom tersebut IDENTITY NOT FOR REPLICATION .	1 = TRUE 0 = FALSE NULL = Invalid input
IsIndexable	Untuk mengecek apakah kolom tersebut <i>Indexable</i> .	1 = TRUE 0 = FALSE NULL = Invalid input
IsOutParam	Untuk mengecek apakah parameter dari prosedur tersebut merupakan <i>output parameter</i> .	1 = TRUE 0 = FALSE NULL = Invalid input
IsPrecise	Untuk mengecek apakah kolom tersebut memiliki nilai presisi. Hanya dapat diterapkan pada <i>deterministic columns</i> .	1 = TRUE 0 = FALSE NULL = Invalid input (bukan deterministic column)
IsRowGuidCol	Untuk mengecek apakah kolom tersebut memiliki tipe data uniqueidentifier dan ditentukan dengan property ROWGUIDCOL.	1 = TRUE 0 = FALSE NULL = Invalid input
Precision	Presisi untuk tiap tipe data kolom atau parameter dari prosedur	Presisi dari tipe data NULL = Invalid input
Scale	Nilai skala untuk kolom atau parameter dari prosedur	Nilai skala NULL = Invalid input

c. Tabel OpenSchema() Microsoft Access

Parameter	Column Name
OpenSchema(adSchemaTables)	<ul style="list-style-type: none"> - table_catalog - table_schema - table_name - table_type - table_guid - decode programion - table_propid - date_created - date_modified
OpenSchema(adSchemaColumns)	<ul style="list-style-type: none"> - table_catalog - table_schema

	<ul style="list-style-type: none">- update_rule- delete_rule- pk_name- fk_name- deferrability
--	---